

NASA Contractor Report 195468

Task 7—Endwall Treatment Inlet Flow Distortion Analysis Final Report

E.J. Hall, D.A. Topp, N.J. Heidegger, G.S. McNulty,
K.F. Weber, and R.A. Delaney
Allison Engine Company
Indianapolis, Indiana

May 1996

Prepared for
Lewis Research Center
Under Contract NAS3-25270



National Aeronautics and
Space Administration

Preface

This report was prepared by Edward J. Hall, David A. Topp, Nathan J. Heidegger, G. Scott McNulty, Kurt F. Weber, and Robert A. Delaney of the Allison Engine Company, Indianapolis, IN. The work was performed under NASA Contract NAS3-25270 from February, 1993 to April, 1995. Principal investigator for this program was Edward J. Hall. The Allison Program Manager for this contract was Robert A. Delaney. The NASA Project Manager was Christopher J. Miller.

Acknowledgements

The authors would like to express their appreciation to the following people who contributed to this program:

Dr. Christopher J. Miller, NASA Lewis Research Center, for his suggestions and critical review of the program

Dr. Andrea Arnone, University of Florence, for his assistance with the development of the multigrid and implicit solution algorithms, and for the use of the codes *TRAF3D*, *JERRYC*, and *TOMC*

N. Duane Melson, NASA Langley Research Center, for his cooperative discussions concerning the iterative implicit flow algorithm

Dr. Ed Greitzer, Massachusetts Institute of Technology, and Dr. Bud Lakshminarayana, Pennsylvania State University, for their insight and suggestions concerning endwall treatment flowfields

Dr. Ali Ameri, NASA Lewis Research Center, and Dr. Okey Kwon, Allison, for their helpful discussions concerning turbulence modeling

Angela Quealy, Dale Hubler, and Scott Townsend, NASA Lewis Research Center, and Helen Wang, Hassan Akay, and Akin Ecer, Indiana University-Purdue University at Indianapolis, for their assistance in developing the parallel computer flow solver

The members of the NASA Lewis Research Center GVIS Lab, for their assistance in producing the animations of the 3-D computational fluid dynamics solutions

Contents

1	SUMMARY	1
2	INTRODUCTION	3
2.1	Description of Compressor Stability and Casing Treatment	3
2.2	Description of Compressor Stability and Inlet Distortion	7
2.3	Objectives of the Present Study	9
3	ADPAC07 NAVIER-STOKES NUMERICAL ALGORITHM	13
3.1	Nondimensionalization	13
3.2	Governing Equations	14
3.2.1	Vector Form of Navier-Stokes Equations	14
3.2.2	Reynolds-Averaged Form of Navier-Stokes Equations	15
3.2.3	Governing Equations for Cartesian Solution	16
3.2.4	Governing Equations for Cylindrical Coordinate Solution	19
3.3	Fluid Properties	20
3.4	Numerical Formulation	21
3.4.1	Finite Volume Discretization	21
3.4.2	Runge-Kutta Time Integration	27
3.4.3	Dissipation Function	31
3.4.4	Implicit Residual Smoothing	32
3.4.5	Multigrid Convergence Acceleration	34
3.4.6	Implicit Time-Marching Algorithm Procedure	36
3.5	Boundary Conditions	38
3.5.1	Standard Inflow/Outflow Boundary Procedures	40
3.5.2	Solid Surface Boundary Procedures	40
3.5.3	Interblock Communication Boundary Procedures	41
3.5.4	Non-Reflecting Inflow/Outflow Boundary Condition Procedures	41
3.6	Turbulence Model	55
3.6.1	Algebraic (Baldwin-Lomax) Turbulence Model	57
3.6.2	Wall Functions	59
3.6.3	Two Equation Turbulence Model	61
3.7	Overall ADPAC07 Numerical Solution Procedure	64
4	Code Parallelization	65
4.1	Introduction	65
4.2	Parallelization Strategy	65
4.3	Description of the ADPAC07 Parallel Implementation	66
4.4	Domain Decomposition	69

4.5	<i>SIXPAC</i> (Block Subdivision) Program	69
4.6	<i>BACPAC</i> Block Reconstruction Program	70
4.7	Load Balancing via Block/Processor Assignment	73
4.8	Serial/Parallel Solution Sequence	74
4.9	Performance of the <i>ADPAC07</i> code in Parallel	75
4.10	General Summary of Parallel Computing Experience Using <i>ADPAC07</i>	82
5	ENDWALL TREATMENT ANALYSIS DESCRIPTION	85
5.1	Endwall Treatment Boundary Conditions	85
5.2	Treatment Modeling Mesh Generation	87
5.3	<i>ADPAC</i> Solution Sequence for Endwall Treatment Flowfield Analysis	89
6	<i>ADPAC07</i> VALIDATION TEST CASES	91
6.1	Turbulent Boundary Layer on a Flat Plate	91
6.2	Bachalo and Johnson Transonic Bump Test Case	92
6.3	Mark II Turbine Vane Cascade	94
6.4	VBI Vane Test of Non-Reflecting Boundary Conditions	99
6.5	Vortex Shedding Behind a Circular Cylinder in Crossflow	107
6.6	Penn State Research Compressor Rotor/Stator/Rotor Interaction	119
6.7	Advanced Small Turboshaft Compressor (ASTC) Wall Function Demonstration Calculation	120
6.8	NASA Rotor 67 Serial/Parallel Comparison	122
6.9	Advanced Small Turboshaft Compressor (ASTC) Parallel <i>ADPAC07</i> Rotor/Stator Aerodynamic Interaction Analysis	124
7	STEADY STATE CASING TREATMENT ANALYSIS WITH NO INLET DISTORTION	129
7.1	MIT Stator Vane with Axial Skewed Groove Casing Treatment	129
7.2	NASA Rotor 5 with Axial Skewed Groove Casing Treatment	133
7.3	AE3007 Fan Rotor Treatment Configuration Study	144
7.4	Summary of Steady State No Distortion Predictions	152
8	TIME DEPENDENT CASING TREATMENT ANALYSIS (NO DISTORTION)	153
8.1	NASA Rotor 5 with Axial Skewed Slot Casing Treatment	153
8.2	AE3007 Fan Rotor with Recessed Vane Casing Treatment	155
8.3	Summary of Time Dependent No Distortion Predictions	160
9	STEADY STATE CASING TREATMENT ANALYSIS WITH INLET DISTORTION	163
9.1	Radial Flow Distortion Analysis	163
9.1.1	Introduction	163
9.1.2	Grid Generation	164
9.1.3	Distorted Inlet Profile	165
9.1.4	Solution Procedure	165
9.1.5	<i>ADPAC07</i> Results and Discussion	167
9.1.6	Grid Density Sensitivity	173

10 TIME-DEPENDENT CASING TREATMENT ANALYSIS WITH IN-	
LET DISTORTION	181
10.1 AE3007 Fan Rotor Circumferential Flow Distortion Analysis	181
10.1.1 Mesh Density Study	184
10.1.2 Parallel Performance Study	184
10.1.3 Circumferential Distortion Results	188
10.2 AE3007 Engine Fan Section Inlet Flow Distortion Simulation	193
10.2.1 Mesh Generation	193
10.2.2 Mesh Density/Flow Condition Study	196
10.2.3 Results	196
11 CONCLUSIONS	201
A ADPAC07 Utility Program Developments	209
A.1 <i>PATCHFINDER</i> Utility Program	209

List of Figures

2.1	Sample compressor operating characteristic at constant speed.	4
2.2	Groove/slot casing treatment geometric descriptions.	5
2.3	Recessed vane casing treatment geometric description.	6
2.4	Effect of inlet distortion on axial compressor performance and stability (nine stage compressor).	7
2.5	Effect of varying distortion sector angle on compressor stall margin.	8
2.6	Effect of dividing distortion sector angle on compressor stall margin.	9
2.7	Illustration of active response of inlet flow distortion to compressor (dashed line illustrates deviation due to effect of compressor on local distortion). . .	10
3.1	<i>ADPAC07</i> Cartesian coordinate system reference.	17
3.2	<i>ADPAC07</i> 2-D single block mesh structure illustration.	22
3.3	<i>ADPAC07</i> 2-D two block mesh structure illustration.	23
3.4	<i>ADPAC07</i> 2-D multiple block mesh structure illustration.	24
3.5	Three-dimensional finite volume cell.	26
3.6	<i>ADPAC07</i> finite volume cell centered data configuration and convective flux evaluation process.	28
3.7	<i>ADPAC07</i> finite volume cell centered data configuration and diffusive flux evaluation process.	29
3.8	Multigrid mesh coarsening strategy and mesh index relation.	34
3.9	Multigrid V cycle strategy.	35
3.10	2-D mesh block phantom cell representation.	39
3.11	<i>ADPAC07</i> contiguous mesh block coupling scheme.	42
3.12	Time history of pressure traces for a 1-D pressure wave traveling in a Mach 0.00 (stationary) flow, demonstrating the advantages of non-reflecting boundary conditions.	45
3.13	Time history of pressure traces for a 1-D pressure wave traveling in a Mach 0.67 flow, demonstrating the advantages of non-reflecting boundary conditions.	46
3.14	Comparison of <i>ADPAC07</i> results for an oscillating flat plate cascade with the Smith linear theory examining the effect of shortening the distance to the computational boundaries and the effect of a non-reflecting boundary condition on the shortest mesh.	47
3.15	Static pressure contours at the casing for Rotor 67 on the long grid using the original turbomachinery boundary conditions.	55
3.16	Static pressure contours at the casing for Rotor 67 on the short grid using the original turbomachinery boundary conditions.	56
3.17	Static pressure contours at the casing for Rotor 67 on the short grid using the new non-reflecting boundary conditions.	56

3.18	Near-wall computational structure for wall function turbulence model. . . .	60
4.1	Improved communication scheme reduces memory requirements.	68
4.2	Boundary condition generation between neighboring blocks often results in large numbers of patches.	71
4.3	Careful block division can preserve levels of multigrid.	72
4.4	Comparison of CPU times for a problem subdivided over a range of processors.	77
4.5	Comparison of speed factors for a problem subdivided over a range of pro- cessors.	79
4.6	Comparison of speed factors for larger problems run on more processors. . .	80
4.7	Comparison of CPU times for larger problems run on more processors. . . .	81
4.8	Comparison of speed factors for a single problem run on more RS6000 pro- cessors.	83
5.1	Endwall treatment numerical boundary conditions.	86
5.2	Endwall treatment mesh generation using <i>TIGG3D</i>	88
5.3	ADPAC sample iteration/mass flow rate history for a constant speed oper- ating characteristic prediction.	90
6.1	Flat plate turbulent boundary layer velocity profile comparison.	92
6.2	Flat plate turbulent boundary layer surface skin friction coefficient comparison.	93
6.3	Bachalo and Johnson test case 161x73 2-D algebraic mesh.	95
6.4	Comparison of predicted and experimental surface static pressure ratio for Bachalo and Johnson transonic axisymmetric bump test case ($M=0.875$). . .	95
6.5	Comparison of predicted algebraic model (upper) and 2-equation model (lower) mach number contours for Bachalo and Johnson transonic axisymmetric bump test case ($M=0.875$).	96
6.6	193x33 2-D mesh system for Mark II turbine vane cascade.	98
6.7	Comparison of experimental and predicted airfoil surface static to inlet total pressure ratio for Mark II turbine vane cascade.	100
6.8	Comparison of experimental and predicted airfoil surface heat transfer coef- ficient for Mark II turbine vane cascade.	101
6.9	VBI vane short grid and long grid boundary outlines and vane exit data plane location.	102
6.10	Exit plane Mach number contours for the VBI vane for various grid lengths and boundary condition combinations.	103
6.11	Inviscid vane surface static pressure distributions along the midspan of the VBI vane.	104
6.12	Comparison of inviscid and viscous predicted midspan vane surface static pressure distributions for VBI vane.	105
6.13	Comparison of midspan blade to blade plane and exit plane (1/4 chord aft of trailing edge) predicted Mach contours for the VBI vane.	106
6.14	Comparison of midspan blade to blade plane and exit plane (1/4 chord aft of trailing edge) predicted static pressure ratio contours for the VBI vane. .	106
6.15	Schematic illustration of vortex shedding behind a circular cylinder in cross- flow test case (Reynolds number = 200.0).	107
6.16	161x65 polar mesh for 2-D vortex shedding behind a circular cylinder in crossflow test case (Reynolds number = 200.0).	108

6.17	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, no multigrid, 40 time steps per oscillation cycle).	109
6.18	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, no multigrid, 80 time steps per oscillation cycle).	110
6.19	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, no multigrid, 160 time steps per oscillation cycle).	111
6.20	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, multigrid, 40 time steps per oscillation cycle).	112
6.21	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, multigrid, 80 time steps per oscillation cycle).	113
6.22	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, multigrid, 160 time steps per oscillation cycle).	114
6.23	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, no multigrid, 40 time steps per oscillation cycle, highly converged).	115
6.24	Predicted lift and drag coefficient time histories for circular cylinder in cross-flow test case (Reynolds number = 200.0, multigrid, 40 time steps per oscillation cycle, highly converged).	116
6.25	Predicted instantaneous streamlines for vortex shedding behind a circular cylinder (Reynolds number = 200.0).	117
6.26	Hybrid O-H mesh system for Penn State research compressor rotor/stator/rotor interaction analysis.	120
6.27	Penn State research compressor rotor/stator/rotor interaction analysis - predicted instantaneous entropy contours.	121
6.28	<i>ADPAC07</i> predicted surface static pressure contours for ASTC compressor.	123
6.29	Serial and parallel solutions of Rotor 67 converge to the same answer.	125
6.30	Single block and multiple block solutions for Rotor 67, run on a serial computer, differ slightly.	126
6.31	Predicted velocity magnitude contours for ASTC compressor <i>ADPAC07</i> rotor/stator aerodynamic interaction analysis.	127
7.1	MIT compressor research rig illustrating stator hub treatment geometry.	130
7.2	MIT stator and axial skewed slot endwall treatment geometry.	130
7.3	MIT stator multiple block mesh system.	132
7.4	MIT stator multiple block mesh system illustrating skewed axial slot endwall treatment mesh.	133
7.5	Comparison of predicted and experimental inlet and exit radial total pressure coefficient profiles for non-treated MIT stator.	134
7.6	Comparison of predicted and experimental inlet and exit radial total pressure coefficient profiles for treated MIT stator.	135
7.7	Predicted flow visualization of MIT stator with smooth endwall.	135
7.8	Predicted flow visualization of MIT stator with treated endwall.	136

7.9	NASA Rotor 5 geometry and design performance summary.	136
7.10	NASA Rotor 5 skewed axial slot endwall treatment geometry.	137
7.11	NASA Rotor 5 skewed axial slot endwall treatment geometry meridional and blade to blade mesh system.	137
7.12	NASA Rotor 5 skewed axial slot endwall treatment geometry axial plane mesh system.	138
7.13	Comparison of predicted and experimental 100% speed pressure ratio ver- sus mass flow operating characteristic for NASA Rotor 5 with and without skewed axial slot endwall treatment.	139
7.14	Comparison of predicted and experimental 100% speed efficiency versus mass flow operating characteristic for NASA Rotor 5 with and without skewed axial slot endwall treatment.	140
7.15	Comparison of predicted and experimental radial total temperature distribu- tions for NASA Rotor 5 with and without skewed axial slot endwall treatment.	141
7.16	Comparison of predicted and experimental radial total pressure distributions for NASA Rotor 5 with and without skewed axial slot endwall treatment.	142
7.17	Comparison of predicted and experimental 100% speed total pressure ratio and efficiency versus mass flow operating characteristic for AE3007 fan with circumferential groove casing treatment.	145
7.18	Comparison of predicted and experimental 100% speed total pressure ratio and efficiency versus mass flow operating characteristic for AE3007 fan with axial slot casing treatment.	146
7.19	Comparison of predicted and experimental 100% speed total pressure ratio and efficiency versus mass flow operating characteristic for AE3007 fan with blade-angle slot casing treatment.	146
7.20	Comparison of predicted and experimental 100% speed total pressure ratio versus mass flow operating characteristic for AE3007 fan with recessed vane casing treatment.	147
7.21	Predicted clearance flow particle traces and rotor surface static pressure con- tours for AE3007 fan with solid endwall (no treatment).	148
7.22	Predicted clearance flow particle traces and rotor surface static pressure con- tours for AE3007 fan with circumferential groove endwall treatment.	148
7.23	Predicted clearance flow particle traces and rotor surface static pressure con- tours for AE3007 fan with axial slot endwall treatment.	149
7.24	Predicted clearance flow particle traces and rotor surface static pressure con- tours for AE3007 fan with blade-angle slot endwall treatment.	149
7.25	Predicted clearance flow particle traces and rotor surface static pressure con- tours for AE3007 fan with recessed vane endwall treatment.	150
7.26	AE3007 fan treatment configuration study summary of axisymmetric aver- aged endwall flow behavior.	151
8.1	NASA Rotor 5 mesh summary illustrating intermediate mesh used to couple blade passage and treatment passage mesh systems.	154
8.2	Predicted instantaneous static pressure contours (axial plane) and static pres- sure contour lines (radial plane) for NASA Rotor 5 rotor/endwall treatment aerodynamic interaction analysis.	156
8.3	Predicted instantaneous treatment slot velocity vector patterns for NASA Rotor 5 rotor/endwall treatment aerodynamic interaction analysis.	157

8.4	Comparison of predicted near rotor tip blade loading for baseline (no treatment) and treated (time-average of rotor/treatment interaction) solutions. .	158
8.5	Comparison of time-dependent predicted and experimental 100% speed total pressure ratio versus mass flow operating characteristic for AE3007 fan with recessed vane casing treatment.	159
8.6	Predicted instantaneous static pressure contours for AE3007 fan rotor with recessed vane casing treatment (45% axial chord).	160
9.1	Comparison of the standard (no groove) casing with the 5 groove endwall casing treatment geometries.	164
9.2	Meridional projection of H-type mesh for AE3007 fan rotor with circumferential groove endwall treatment used for radial distortion study.	166
9.3	Total pressure inlet profiles.	167
9.4	Pressure ratio and efficiency for the AE3007 fan rotor.	168
9.5	Comparison of predicted and experimental radial distributions of total pressure ratio, total temperature ratio, efficiency, and swirl velocity for the AE3007 fan both with and without circumferential groove endwall treatment, and with and without tip radial distortion at peak efficiency.	169
9.6	Comparison of predicted and experimental radial distributions of total pressure ratio, total temperature ratio, efficiency, and swirl velocity for the AE3007 fan both with and without circumferential groove endwall treatment, and with and without tip radial distortion at near-stall.	170
9.7	Relative Mach number contours located at the tip radial plane at peak efficiency and no inlet distortion for both the standard casing (upper) and endwall treated (lower) configurations.	171
9.8	Relative Mach number contours located at the tip radial plane at near-stall and no inlet distortion for both the standard casing (upper) and endwall treated (lower) configurations.	172
9.9	Relative Mach number contours located at the tip radial plane at peak efficiency with tip radial distortion for both the standard casing (upper) and endwall treated (lower) configurations.	172
9.10	Relative Mach number contours located at the tip radial plane at near-stall with tip radial distortion for both the standard casing (upper) and endwall treated (lower) configurations.	173
9.11	Circumferentially-averaged velocity vectors (shaded by pressure) for peak efficiency operating point without distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span. . . .	174
9.12	Circumferentially-averaged velocity vectors (shaded by pressure) for near-stall operating point without distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span. . . .	175
9.13	Circumferentially-averaged velocity vectors (shaded by pressure) for peak efficiency operating point with inlet distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span. . . .	176
9.14	Circumferentially-averaged velocity vectors (shaded by pressure) for near-stall operating point with inlet distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span. . . .	177

9.15	Pressure ratio and efficiency speedlines (100% corrected speed) showing comparison between two grid densities. Filled symbols represent <i>fine</i> grid solutions, open symbols represent <i>coarse</i> grid solutions.	178
10.1	Multiple-block mesh system for AE3007 circumferential inlet flow distortion study.	183
10.2	Comparison of predicted and experimental pressure ratio versus mass flow characteristic for the AE3007 fan (100% corrected speed).	185
10.3	Comparison of predicted and experimental efficiency versus mass flow characteristic for the AE3007 fan (100% corrected speed).	186
10.4	Inflow and outflow axial total pressure contour pattern for AE3007 circumferential distortion calculation (untreated rotor, near stall operating point).	189
10.5	Inflow and outflow axial plane total pressure contour pattern for AE3007 circumferential distortion calculation (treated rotor, near stall operating point).	190
10.6	Instantaneous predicted radial distribution of circumferentially averaged total pressure profiles for several blade passages of the AE3007 fan rotor (untreated) with circumferential inlet distortion.	191
10.7	AE3007 engine flowpath and <i>TIGG3D</i> block boundary illustration.	194
10.8	AE3007 engine flowpath and <i>ADPAC07</i> block numbering illustration.	195
10.9	Predicted surface static pressure countours and near surface streamlines for the AE3007 fan (Mach=0.2, angle of attack=20.0 degrees).	196
10.10	Predicted instantaneous surface static pressure contours for AE3007 engine simulation (Mach=0.2, angle of attack=30.0 degrees).	197
10.11	Predicted instantaneous near fan face inlet total pressure contours for the AE3007 engine simulation (Mach=0.2, angle of attack=30.0 degrees).	198
10.12	Predicted instantaneous near fan face inlet total pressure contours for the AE3007 engine simulation (Mach=0.2, angle of attack=30.0 degrees).	199
A.1	Approximate wall clock run times for various <i>PATCHFINDER</i> test case configurations run on a Silicon Graphics 4D-35 workstation (5 MFLOP).	210

List of Tables

6.1	Mark II vane cascade design parameters.	97
6.2	Summary of <i>ADPAC'07</i> lift and drag coefficient and Strouhal number predictions for vortex shedding behind a circular cylinder in crossflow (Reynolds number = 200.0).	118
6.3	Summary of <i>ADPAC'07</i> CPU times for vortex shedding behind a circular cylinder in crossflow (Reynolds number = 200.0).	119
6.4	Comparison of predicted performance figures for ASTC compressor.	122
10.1	Summary of <i>ADPAC'07</i> parallel CPU performance on LACE SP1 cluster for circumferential distortion study untreated rotor mesh.	187
10.2	Summary of flight conditions for <i>ADPAC'07</i> engine/fan section distortion calculation.	193

NOTATION

A list of the symbols used throughout this document and their definitions is provided below for convenience.

Roman Symbols

a . . . speed of sound
 c_f . . . skin friction coefficient
 c_p . . . gas specific heat at constant pressure
 c_v . . . gas specific heat at constant volume
 e . . . total internal energy
 i . . . first grid index of numerical solution
 j . . . second grid index of numerical solution
 k . . . third grid index of numerical solution or thermal conductivity
 k . . . turbulent kinetic energy
 l . . . Van Driest damping function
 n . . . rotational speed (revolutions per second) or time step level
 p . . . pressure
 r . . . radius or radial coordinate
 t . . . time
 u_x . . . velocity in the Cartesian coordinate system x direction
 u_y . . . velocity in the Cartesian coordinate system y direction
 u_z . . . velocity in the Cartesian coordinate system z direction

v_r ... velocity in the cylindrical coordinate system radial direction
 v_θ ... velocity in the cylindrical coordinate system circumferential direction
 w_{rel} ... relative velocity in the circumferential direction ($= v_\theta - r\omega$)
 x ... Cartesian coordinate system coordinate
 y ... Cartesian coordinate system coordinate
 z ... Cartesian coordinate system coordinate
 A^+ ... turbulence model constant
ADPAC07 ... Advanced Ducted Propfan Analysis Code Version 07
ADSPIN ... ADPAC post processing program
ASCII ... American Standard Code for Information Interchange
CFL ... Courant-Freidrichs-Lewy number ($\Delta t / \Delta t_{max, stable}$)
 D ... diameter
 F ... i coordinate direction flux vector
 G ... j coordinate direction flux vector
GRIDGEN ... Multiple block general purpose mesh generation system
GROOVY ... Treatment groove mesh generation program
 H ... k coordinate direction flux vector
 H_{total} ... total enthalpy
 J ... advance ratio ($J = U/nD$)
JERRYC ... TRAF2D Airfoil Cascade C-Mesh Generation Program
 K ... cylindrical coordinate system source vector
 L ... reference length
 M ... Mach number
MFLOP ... million floating point operations per second (CPU speed factor)
 N ... Number of blades
 Q ... vector of conserved variables
 R ... gas constant or residual or maximum radius
 \mathcal{R} ... turbulent Reynolds number
 Re ... Reynolds Number
 P ... turbulence kinetic energy production term
 Pr ... gas Prandtl Number
 S ... surface area normal vector
SDBLIB ... Scientific DataBase Library (binary file I/O routines)
 T ... Temperature
TRAF2D ... TRAF2D Navier-Stokes analysis code
TRAF3D ... TRAF3D Navier-Stokes analysis code
TOMC ... TRAF2D Airfoil Cascade C-Mesh Generation Program
 U ... Freestream velocity (units of length/time)
 V ... volume

Greek Symbols

γ ... specific heat ratio
 Δ ... calculation increment
 ϵ ... turbulence dissipation parameter
 ∇ ... gradient vector operator
 ω ... vorticity
 ρ ... density

μ ... coefficient of viscosity
 τ ... fictitious time or shear stress
 $\Pi_{i,j}$... fluid stress tensor

Subscripts

[]₁ ... inlet value
[]₂ ... exit value
[]_{ax} ... pertaining to the axial (x) cylindrical coordinate
[]_{coarse} ... coarse mesh value
[]_{effective} ... effective value
[]_{fine} ... fine mesh value
[]_{freestream} ... freestream value
[] _{i,j,k} ... grid point index of variable
[]_{laminar} ... laminar flow value
[]_{max} ... maximum value
[]_{min} ... minimum value
[]_{nearwall} ... near wall value
[]_{non-dimensional} ... non-dimensional value
[]_r ... pertaining to the radial (r) cylindrical coordinate
[]_{ref} ... reference value
[]_{stable} ... value implied by linear stability
[]_t ... turbulent flow value
[]_{total} ... total (stagnation) value
[]_{turbulent} ... turbulent flow value
[]_{wall} ... value at the wall
[]_x ... pertaining to the x Cartesian coordinate
[]_y ... pertaining to the y Cartesian coordinate
[]_z ... pertaining to the z Cartesian coordinate
[] _{θ} ... pertaining to the circumferential (θ) cylindrical coordinate

Superscripts

[]⁺ ... Turbulent velocity profile coordinate
[]^{*} ... Intermediate value
[] ^{n} ... Time step index
[] ... (no overscore) nondimensional variable
[] ... Dimensional variable
[] ... Time-averaged variable
[] ... Density-weighted time-averaged variable
[] ... Vector variable
[] ... Tensor variable

Chapter 1

SUMMARY

The overall objective of this study was to develop a 3-D numerical analysis for compressor casing treatment flowfields, and to perform a series of detailed numerical predictions to assess the effectiveness of various endwall treatments for enhancing the efficiency and stall margin of modern high speed fan rotors. Particular attention was given to examining the effectiveness of endwall treatments to counter the undesirable effects of inflow distortion. The motivation behind this study was the relative lack of physical understanding of the mechanics associated with the effects of endwall treatments and the availability of detailed computational fluid dynamics (CFD) codes which might be utilized to gain a better understanding of these flows.

Calculations were performed using a cylindrical coordinate system utilizing three different gridding techniques based on the type of casing treatment being tested and the level of complexity desired in the analysis. The interface between the casing treatment flow and the primary rotor flowpath flow was addressed using either a direct coupled approach, a time-averaged approach, or a time-accurate approach. In each case, the casing treatment itself is modeled as a discrete object in the overall analysis, and the flow through the casing treatment is determined as part of the solution.

A series of calculations was performed for both treated and untreated modern fan rotors both with and without inflow distortion. The effectiveness of the various treatments were quantified, and several physical mechanisms by which the effectiveness of endwall treatments was achieved are discussed. This report represents the cumulative efforts of Tasks 6 and 7 under NASA Contract NAS3-25270.

Chapter 2

INTRODUCTION

2.1 Description of Compressor Stability and Casing Treatment

Compressor aerodynamic stability is a major concern for manufacturers of gas turbine aircraft engines and industrial power plants. For a given shaft rotational speed, the compressor mass flow rate versus the overall total pressure rise characteristic (often referred to as a constant speed line) is commonly used as a measure for evaluating operating stability. An illustration of a constant speed operating characteristic is given in Figure 2.1. Improving the flow margin between the compressor operating point and the stall-limit point will, in general, improve the useful operating range of the engine. Compressors are normally designed with a significant excess flow margin to compensate for factors, which, over time, can degrade compressor performance and ultimately limit the stable operational domain of the compressor. Under many conditions, unstable flow conditions are initiated in the endwall flow regions of a compressor. If aerodynamic stall in the endwall region can be delayed, the weight flow range (and hence, stability) of the compressor may be increased.

Experimental data [1],[2], indicate that compressor endwall treatments such as grooves, slots (see Fig 2.2), and/or recessed vane sets (see Fig 2.3) can effectively delay stall and increase the weight flow range of a compressor. Prince et. al. [3] and Fujita and Takata [4] examined many types of grooves over a low speed rotor. Flow features in treatment grooves and in the rotor relative flow field were presented by Smith and Cumpsty [6]. A similar treatment (axially skewed grooves) was successfully applied under a stator vane row (Cheng et al. [7]) and details of the vane passage flow field were illuminated by Johnson and Greitzer [8]. With some features of the flow field known, mechanisms for the stall margin improvement were postulated. Flow injection (Takata and Tsukuda [5]) from the grooves and/or flow withdrawal into the grooves (Johnson [8]) have been suggested as potential mechanisms by which blade row stall is delayed by endwall treatments. A well planned experiment by Lee and Greitzer [9] showed both flow suction and injection from the treatment produced a stall margin improvement.

Another form of casing treatment employs a large cavity upstream and over part of the rotor tips. The cavity often has vanes embedded in it and such a “recess vaned” casing treatment is shown in Fig 2.3. This treatment [10] has been applied to ventilation fans and significantly increases the stable flow range. Miyake et al. [11] and Azimian et al. [12] [13] have evaluated various characteristics of this treatment and have shown the general flow structure in the cavity and around the rotor.

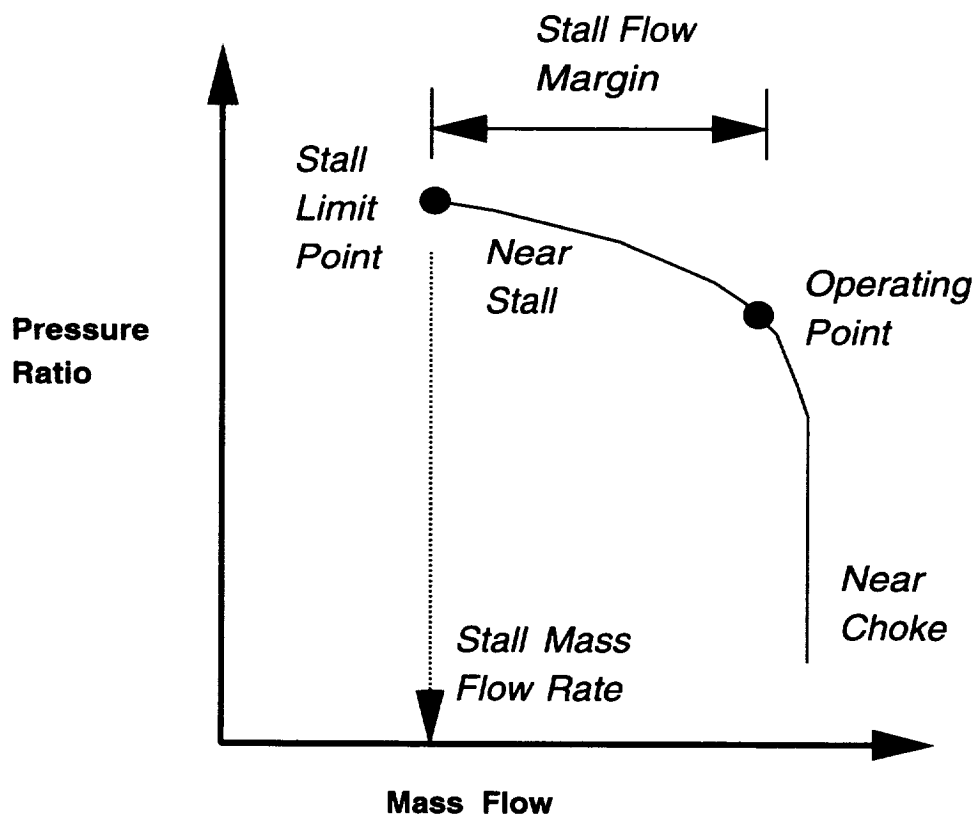


Figure 2.1: Sample compressor operating characteristic at constant speed.

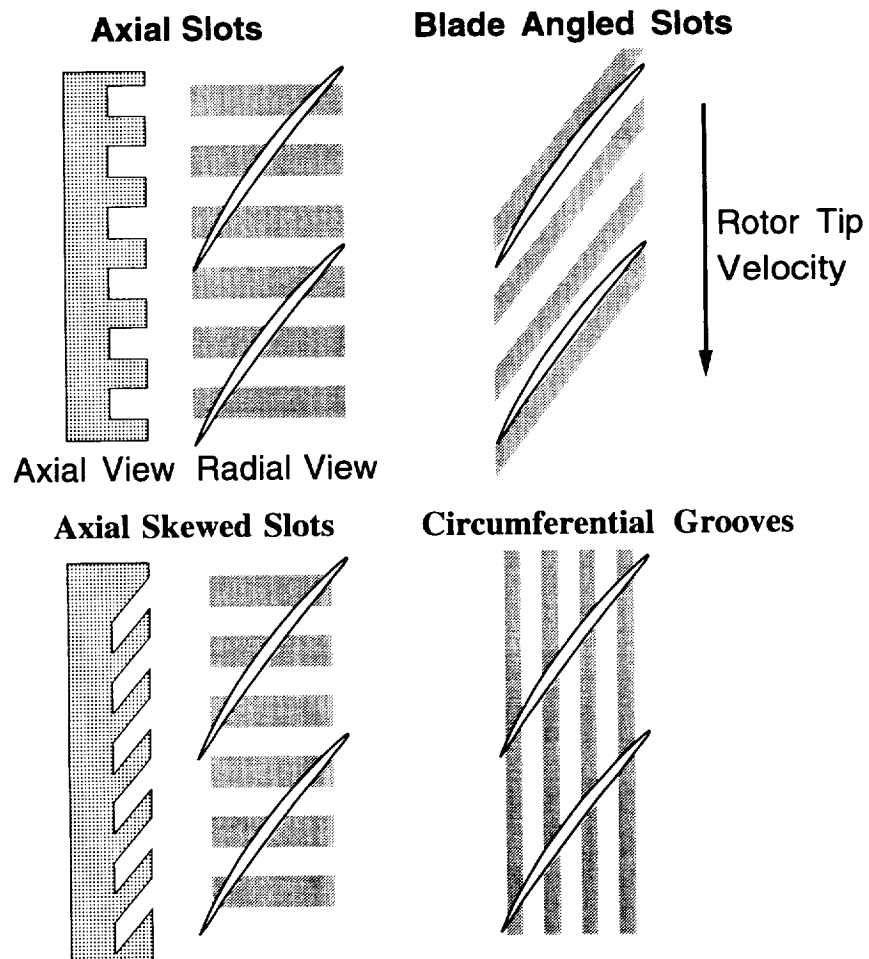


Figure 2.2: Groove/slot casing treatment geometric descriptions.

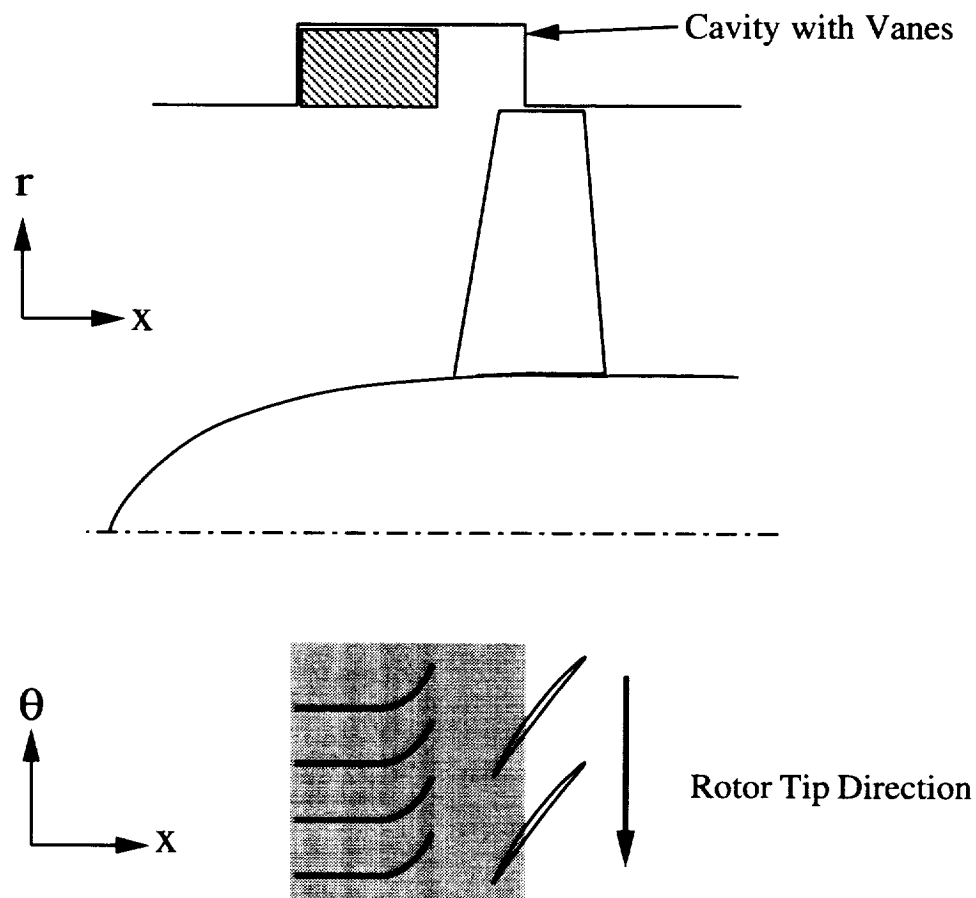


Figure 2.3: Recessed vane casing treatment geometric description.

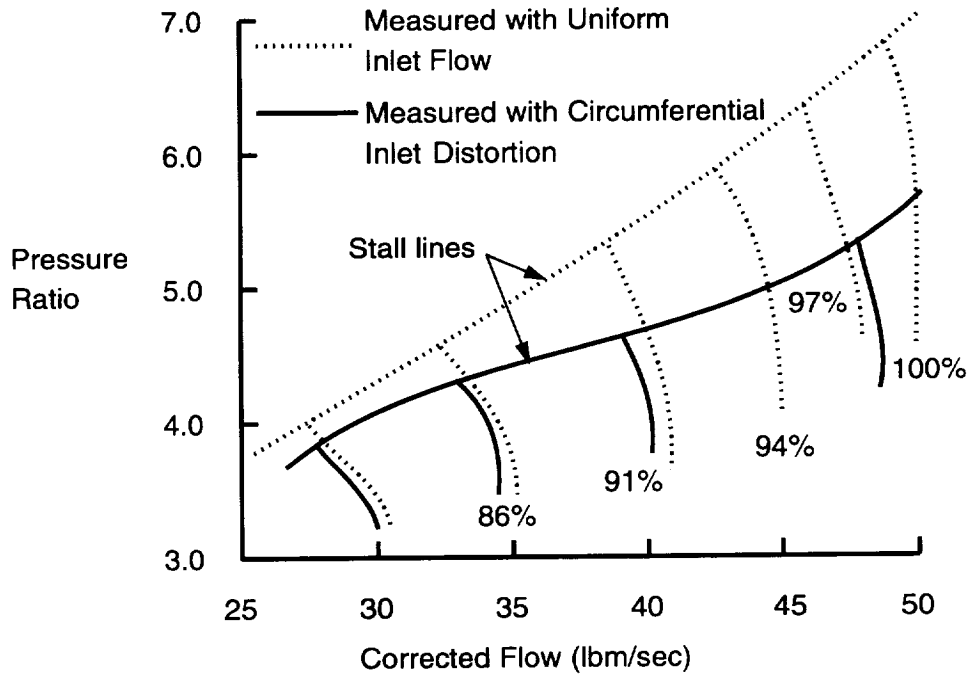


Figure 2.4: Effect of inlet distortion on axial compressor performance and stability (nine stage compressor).

The research to date has uncovered many flow features of casing treatment, but the exact mechanism of the stall suppression and the relative effect on performance imposed by endwall treatments are not fully understood. Computational fluid analysis has been a reasonable predictor of turbomachinery flow fields, and can be seen as a tool to aid in understanding the complexities of casing treatment flow phenomena. Previous attempts to compute compressor casing treatment flow fields are few in number, and have generally specified the resultant flow through the treatment region based on inference from experimental data [14].

2.2 Description of Compressor Stability and Inlet Distortion

It is well known that a compressor's stable operating range is seriously degraded in the presence of inlet flow non-uniformities (i.e. distortion). In an aircraft application, engine stability problems can occur during rapid maneuvers or in strong cross-winds due to the total pressure non-uniformities that are created at the compressor inlet. Figure 2.4 illustrates the substantial decrease in compressor stall margin that was measured for a nine stage axial compressor with a circumferential inlet distortion [15].

Inlet distortion patterns occurring in aircraft systems are generally non-uniform in both the circumferential and radial directions. To simplify testing, data correlation, and analytical study, these patterns are normally decomposed into 3 separate categories: steady circumferential, steady radial and unsteady distortions [16]. Reasonable success has been made in understanding the effects of inlet distortion by independently considering the steady circumferential (which the rotor sees as an unsteady inlet flow) and radial types.

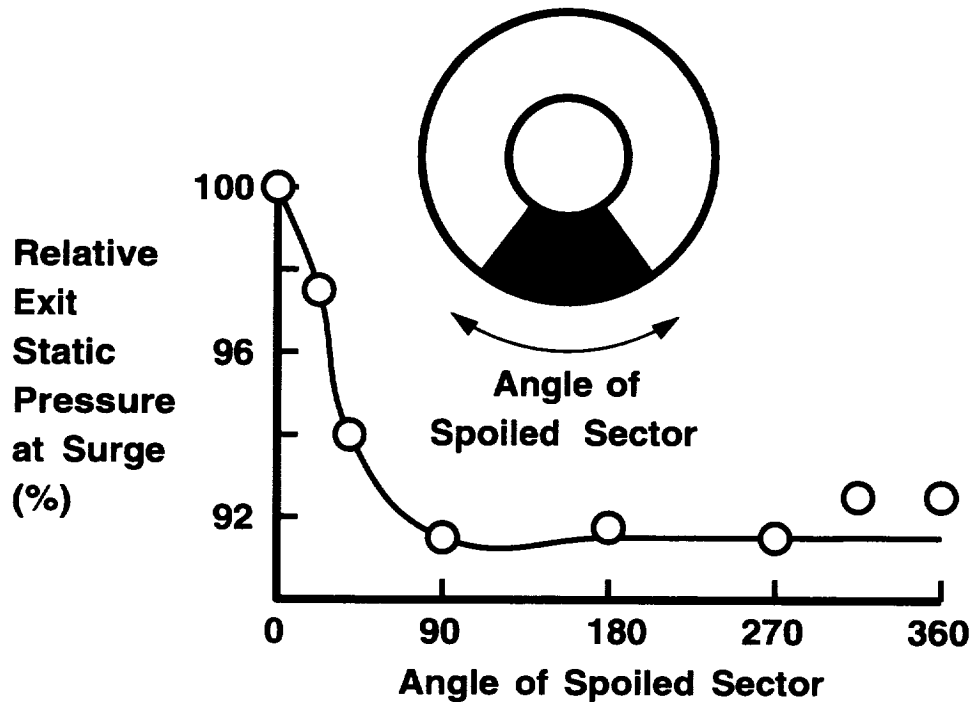


Figure 2.5: Effect of varying distortion sector angle on compressor stall margin.

There are several interesting features of a compressor's response to inlet distortion that have been observed. Figure 2.5 [17] shows the effect of increasing the circumferential extent of the distortion on the compressor exit pressure at surge (as a percent of uniform inlet value). It can be seen that the deleterious effect of the distortion increases until a "critical" angle of about 90 degrees is reached, beyond which no significant additional performance degradation occurs. Figure 2.6 shows the effect of taking the 90 degree distortion and subdividing it into smaller sections such that the sum remains constant. Even though the overall extent is the same for these tests, the distortions with smaller individual sectors show a reduced effect on compressor stability. In other words, distortions with low harmonic content are more harmful than those with higher harmonics. Another important point to understand is the strong interaction that occurs between the compressor and the distorted flow field. As illustrated schematically in Figure 2.7 [17], and discussed in Ref. [16], the compressor plays an active role in determining the velocity distribution that will occur at the compressor face, which is what the individual compressor airfoils respond to. These results indicate the important role that unsteady fluid processes play in determining a compressor's response to a distorted inlet.

Over the years there has been much work towards the development of analytical tools for the prediction of the effects of circumferential inlet distortion on compressor stability. These methods range from correlations of experimental data [18] to simple analytical approaches, such as the well known compressor-in-parallel model [19]. These simple methods are routinely used in practice due to their demonstrated ability to predict experimental trends. A significant drawback is the costly amount of empirical data required as input. There has been some work to develop tools that actually model the important unsteady physical processes and thus reduce the amount of empirical input [20] [21]. These methods are based on a hydrodynamic (linear) stability analysis of the assumed two-dimensional,

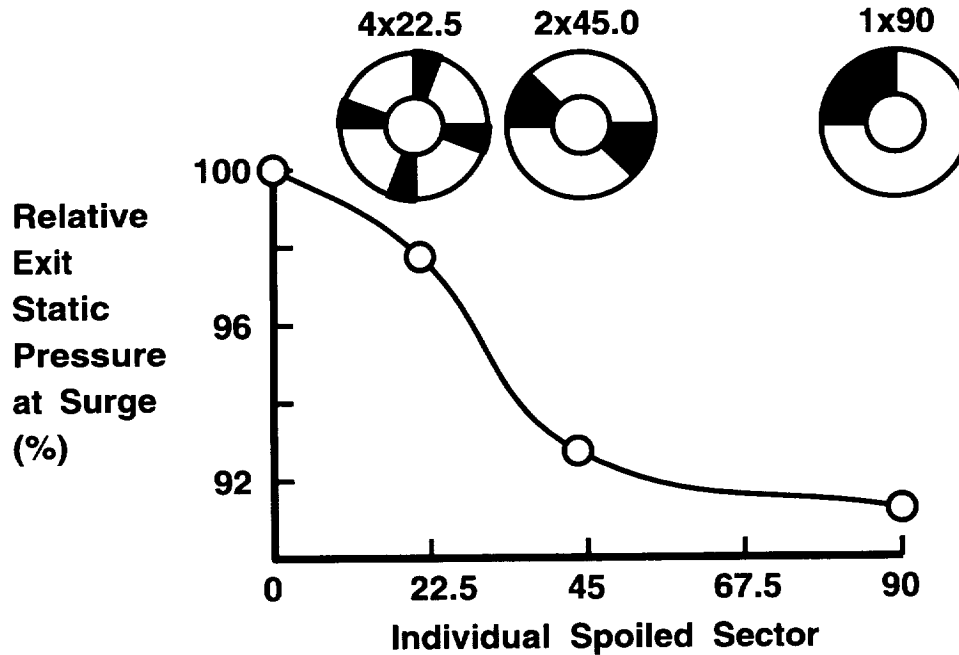


Figure 2.6: Effect of dividing distortion sector angle on compressor stall margin.

incompressible distorted compressor flowfield. This approach has been shown to reproduce known experimental trends, such as the decrease in compressor stability with increasing distortion amplitude and circumferential extent, as well as the prediction of a “critical” angle. However, these models also rely on the input of the steady-state pressure rise versus flow characteristic for the compressor and this is the reason these models have not been widely adopted by industry. The greatest contribution of these methods has been the increased understanding of physical mechanisms that they have provided.

The phenomenal computational speed that is now available using current computational fluid dynamic codes on modern computer systems has opened the door for a new approach to the distortion problem. It is now possible using state-of-the-art (CFD) codes to model the unsteady, three-dimensional viscous compressor flowfield in the presence of a distorted inlet. The basic idea is to use these computations to provide insight into the detailed fluid mechanics of this phenomena. This approach is unique because it allows the detailed three-dimensional flow features to be considered (e.g. tip clearance vortex), as well as the interactions between blade passages without assumptions regarding the response of each passage. The long term goal is to someday be able to predict the effect of a given inlet distortion on a given compressor without the use of expensive rig testing.

2.3 Objectives of the Present Study

The overall objective of this study was to develop a 3-D numerical analysis for compressor casing treatment flowfields, and to perform a series of detailed numerical predictions to assess the effectiveness of various endwall treatments for enhancing the efficiency and stall margin of modern high speed fan rotors. Particular attention was given to examining the effectiveness of endwall treatments to counter the undesirable effects of inflow distortion. The motivation behind this study was the relative lack of physical understanding of the

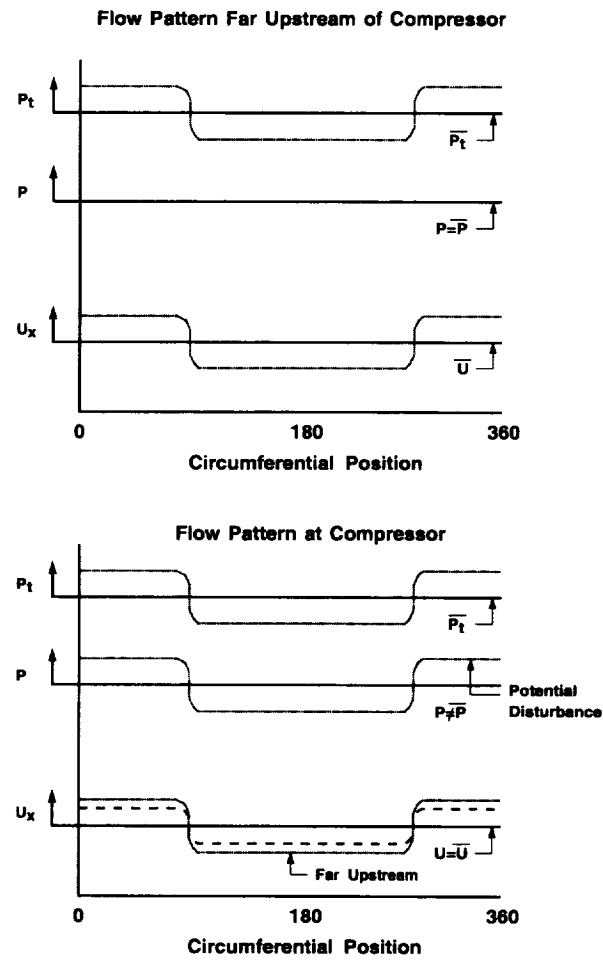


Figure 2.7: Illustration of active response of inlet flow distortion to compressor (dashed line illustrates deviation due to effect of compressor on local distortion).

mechanics associated with the effects of endwall treatments and the availability of detailed computational fluid dynamics (CFD) codes which might be utilized to gain a better understanding of these flows. This study represents one of the first attempts to simultaneously compute the coupled flow through the blade passage and the treatment region for various compressor endwall treatment configurations. The computational tool used for this study was the *ADPAC07* code, a flexible viscous flow aerodynamic tool developed specifically for turbomachinery geometries. The secondary objectives of this study were directed at enhancing the capabilities of the *ADPAC07* code by incorporating computational enhancements such as code parallelization, an implicit time-marching algorithm, and an advanced turbulence model. Each of these enhancements are described in detail in the sections which outline the numerical algorithm.

Chapter 3

ADPAC07 NAVIER-STOKES NUMERICAL ALGORITHM

Predictions for the casing treatment and inlet distortion flows described in this study were obtained using the *ADPAC07* computer program. The *ADPAC07* code is a general purpose turbomachinery aerodynamic design analysis tool which has undergone extensive development, testing, and verification [22], [23], [24], [30]. There is also extensive documentation available for the *ADPAC07* program [26], [27], [29], [28]. Briefly, the *ADPAC07* analysis utilizes a finite volume, multigrid-based Runge-Kutta time-marching solution algorithm to solve a time-dependent form of the 3-D Reynolds-Averaged Navier-Stokes equations. A relatively standard Baldwin-Lomax [34] turbulence model was incorporated to compute the turbulent shear stresses. An advanced two-equation turbulence model was also developed for enhanced turbulent flow predictions. The code employs a multiple-blocked mesh discretization which provides extreme flexibility for analyzing complex geometries. The block gridding technique enables the coupling of complex, multiple-region domains with common grid interface boundaries through specialized boundary condition procedures. The *ADPAC07* analysis has been successfully utilized to predict both the steady state and time-dependent aerodynamic interactions occurring in modern multistage compressors and turbines.

In this chapter, the governing equations and computational model methodology for the *ADPAC07* code are described. In some cases, additional capabilities are available in the *ADPAC07* program, and these are described further in References [24], [29]. The definitions of the pertinent variables used in this chapter may be found in Nomenclature.

3.1 Nondimensionalization

To simplify the implementation of the numerical solution, all variables are nondimensionalized by reference values as follows (note that variables with the caret (e.g. $\hat{\phi}$) are *dimensional* variables and consequently variables *without* a caret (e.g. ϕ) are *nondimensional* variables):

$$\begin{aligned} y &= \frac{\hat{y}}{\hat{L}_{ref}}, & z &= \frac{\hat{z}}{\hat{L}_{ref}}, & v_x &= \frac{\hat{v}_x}{\hat{V}_{ref}}, & v_y &= \frac{\hat{v}_y}{\hat{V}_{ref}}, & v_z &= \frac{\hat{v}_z}{\hat{V}_{ref}} \\ x &= \frac{\hat{x}}{\hat{L}_{ref}}, & r &= \frac{\hat{r}}{\hat{L}_{ref}}, & v_{ax} &= \frac{\hat{v}_{ax}}{\hat{V}_{ref}}, & v_r &= \frac{\hat{v}_r}{\hat{V}_{ref}}, & v_\theta &= \frac{\hat{v}_\theta}{\hat{V}_{ref}} \end{aligned}$$

$$\begin{aligned}
p &= \frac{\hat{p}}{\hat{p}_{ref}}, \quad \mu = \frac{\hat{\mu}}{\hat{\mu}_{ref}}, \quad c_p = \frac{\hat{c}_p}{\hat{R}_{ref}}, \quad c_v = \frac{\hat{c}_v}{\hat{R}_{ref}}, \quad k = \frac{\hat{k}}{\hat{k}_{ref}} \\
T &= \frac{\hat{T}}{\hat{T}_{ref}}, \quad \rho = \frac{\hat{\rho}}{\hat{\rho}_{ref}}, \quad k = \frac{\hat{k}}{\hat{V}_{ref}^2}, \quad \mathcal{R} = \frac{\hat{\mathcal{R}}}{\hat{V}_{ref} \hat{L}_{ref}}.
\end{aligned} \tag{3.1}$$

The reference quantities are defined as follows:

- \hat{L}_{ref} is a constant user-defined length scale
- \hat{p}_{ref} is normally the inlet total pressure (user-defined)
- $\hat{\rho}_{ref}$ is the freestream or inlet total density ($\hat{\rho}_{ref} = \hat{p}_{ref} / \hat{R}_{ref} / \hat{T}_{ref}$)
- \hat{V}_{ref} is determined from the freestream total acoustic velocity as
$$\hat{V}_{ref} = \frac{\hat{a}_{ref}}{\sqrt{\gamma}} = \sqrt{\hat{R}_{ref} \hat{T}_{ref}}$$
- $\hat{\mu}_{ref}$ is determined from the other factors as:
$$\hat{\mu}_{ref} = \hat{\rho}_{ref} \hat{V}_{ref} \hat{L}_{ref}$$
- \hat{k}_{ref} is the freestream thermal conductivity (extracted from user-defined parameters such as γ and Prandtl number)
- \hat{R}_{ref} is the freestream gas constant (user-defined)
- \hat{T}_{ref} is normally the inlet total temperature (user-defined)

3.2 Governing Equations

The *ADPAC07* numerical solution procedure is based on an integral representation of the strong conservation law form of the 3-D Reynolds-averaged Navier-Stokes equations expressed in either a cylindrical or Cartesian coordinate system. User input determines which solution scheme is selected, and can be varied on a block by block basis. The Euler equations may be derived as a subset of the Navier-Stokes equations by neglecting viscous dissipation and thermal conductivity terms (i.e. μ and $k = 0$).

The derivations of the various forms of the equations employed in the *ADPAC07* code are outlined below.

3.2.1 Vector Form of Navier-Stokes Equations

The Navier-Stokes equations may be efficiently described in a coordinate independent vector form as follows (see e.g. [42])

Continuity

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0 \tag{3.2}$$

Momentum

$$\frac{\partial(\rho \vec{V})}{\partial t} + \nabla \cdot \rho \vec{V} \vec{V} = \rho \vec{f} + \nabla \cdot \mathbf{\Pi}_{ij} \tag{3.3}$$

Energy

$$\frac{\partial(\rho \epsilon)}{\partial t} + \nabla \cdot (\rho \epsilon \vec{V}) = \frac{\partial Q}{\partial t} - \nabla \cdot \vec{q} + \rho \vec{f} \cdot \vec{V} + \nabla \cdot (\mathbf{\Pi}_{ij} \cdot \vec{V}) \tag{3.4}$$

Here ρ is density, \vec{V} is the fluid velocity vector, ϵ is the fluid total internal energy, t is time, ∇ is the spatial gradient operator, $\mathbf{\Pi}_{ij}$ is the fluid stress tensor, \vec{f} is an external force vector, Q represents added heat, and \vec{q} is the fluid conduction heat flux vector.

3.2.2 Reynolds-Averaged Form of Navier-Stokes Equations

Direct computation of turbulent flows using the Navier-Stokes equations in the form above is simply not practical at this point, and instead, we assume that the turbulence is *stationary* (see e.g. Wilcox [63]), and can be effectively represented numerically as a time-averaged effect. In this respect, it is useful to derive the Reynolds-averaged form of the Navier-Stokes equations by introducing time averaging operators. Any instantaneous flow variable $f(x, t)$ can be decomposed into a time-averaged and a fluctuating component as

$$f(x, t) = \bar{f}(x) + f'(x, t) \quad (3.5)$$

The time average $\bar{f}(x)$ is defined as

$$\bar{f}(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} f(x, t) dt \quad (3.6)$$

Similarly, for compressible flows, it is useful to define the density weighted time average as

$$\tilde{f}(x, t) = \tilde{f}(x) + f''(x, t) \quad (3.7)$$

where now the density weighted time averaged variable is defined as

$$\tilde{f}(x) = \frac{\overline{\rho f}}{\rho} \quad (3.8)$$

Application of the mass-weighted averaging procedure to the Navier-Stokes equations (see e.g. [42]) yields the Reynolds-averaged Navier-Stokes equations expressed in vector form as

Continuity

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j}(\bar{\rho} \tilde{u}_j) = 0 \quad (3.9)$$

Momentum

$$\frac{\partial}{\partial t}(\bar{\rho} \tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho} \tilde{u}_i \tilde{u}_j) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}(\bar{\tau}_{ij} - \overline{\rho u_i'' u_j''}) \quad (3.10)$$

Energy

$$\frac{\partial}{\partial t}(\bar{\rho} \tilde{H}_{total}) + \frac{\partial}{\partial x_j}(\bar{\rho} \tilde{u}_j \tilde{H}_{total} + \overline{\rho u_j'' H_{total}} - k \frac{\partial \bar{T}}{\partial x_j}) = \frac{\partial \bar{p}}{\partial t} + \frac{\partial}{\partial x_j}(\tilde{u}_i \bar{\tau}_{ij} + \overline{u_i'' \tau_{ij}}) \quad (3.11)$$

where

$$\bar{\tau}_{ij} = \mu \left[\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial \tilde{u}_k}{\partial x_k} \right] + \mu \left[\left(\frac{\partial \overline{u_i''}}{\partial x_j} + \frac{\partial \overline{u_j''}}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial \overline{u_k''}}{\partial x_k} \right] \quad (3.12)$$

where δ_{ij} is the Kronecker delta function ($\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$) and u_i represents the velocity vector components. The complication in this analysis is the presence of terms of the form $\overline{\rho u_i'' u_j''}$. These terms are often referred to as Reynolds stresses, and the specification of these terms is referred to as the turbulent *closure* problem. A large portion of turbulence modeling research is dedicated to suitably closing the system of equations by defining procedures to compute the Reynolds stress terms. In this study, turbulence closure is performed by employing the Boussinesq approximation. Boussinesq [41] suggested that

the apparent turbulent stresses might be related to the mean strain rate through an *eddy* viscosity of the form

$$-\overline{\rho u_i'' u_j''} = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \left(\mu_t \frac{\partial \bar{u}_k}{\partial x_k} + \bar{\rho} \bar{k} \right) \quad (3.13)$$

where \bar{k} is the *kinetic energy of turbulence* defined as $\bar{k} = \overline{u_i'' u_i''}/2$. The resulting simplification is that all Reynolds stress terms are eliminated in favor of a modified viscosity $\mu_{effective} = \mu_{laminar} + \mu_{turbulent}$ where $\mu_{turbulent}$ is the eddy viscosity described above. The turbulent flow thermal conductivity term is also treated as the combination of a laminar and turbulent quantity as

$$k_{effective} = k_{laminar} + k_{turbulent} \quad (3.14)$$

For turbulent flows, the turbulent thermal conductivity $k_{turbulent}$ is determined from a turbulent Prandtl number $Pr_{turbulent}$ such that

$$Pr_{turbulent} = \frac{c_p \mu_{turbulent}}{k_{turbulent}} \quad (3.15)$$

The turbulent Prandtl number is normally chosen to have a value of 0.9. The turbulence models described later in this report define the means by which $\mu_{turbulent}$ is prescribed.

Coordinate dependent forms of the Reynolds-averaged Navier-Stokes equations used in the numerical solution procedures are given in the sections which follow.

3.2.3 Governing Equations for Cartesian Solution

In this section, the governing equations for a Cartesian coordinate system solution are developed. In this discussion, since all solutions for turbulent flow employ the Boussinesq approximation, the overscores denoting time averaged (e.g. $\bar{\rho}$) and density weighted time averaged (e.g. $\bar{\rho} v_x$) have been removed for simplicity.

The Reynolds-averaged Navier-Stokes equations for a Cartesian coordinate system may be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_{inv}}{\partial x} + \frac{\partial G_{inv}}{\partial y} + \frac{\partial H_{inv}}{\partial z} = S + \frac{\partial F_{vis}}{\partial x} + \frac{\partial G_{vis}}{\partial y} + \frac{\partial H_{vis}}{\partial z} \quad (3.16)$$

For a Cartesian solution, the vector of dependent variables Q is defined as

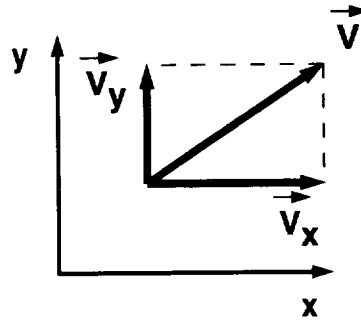
$$Q = \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho e_t \end{bmatrix} \quad (3.17)$$

where the velocity components v_x , v_y , and v_z are the absolute velocity components in the x , y , and z coordinate directions, respectively (see e.g. - Fig. 3.1). The total internal energy is defined as

$$e_t = \frac{p}{(\gamma - 1)\rho} + \frac{1}{2}(v_x^2 + v_y^2 + v_z^2) \quad (3.18)$$

The individual flux functions are defined as

2-D Cartesian Coordinate System Reference



3-D Cartesian Coordinate System Reference

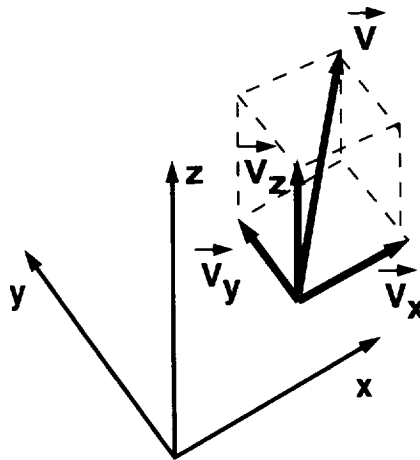


Figure 3.1: *ADPAC07* Cartesian coordinate system reference.

$$F_{inv} = \begin{bmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_y \\ \rho v_x v_z \\ \rho v_x H_{total} \end{bmatrix}, \quad G_{inv} = \begin{bmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + p \\ \rho v_y v_z \\ \rho v_y H_{total} \end{bmatrix}, \quad H_{inv} = \begin{bmatrix} \rho v_z \\ \rho v_x v_z \\ \rho v_y v_z \\ (\rho v_z^2 + p) \\ \rho v_z H_{total} \end{bmatrix} \quad (3.19)$$

$$F_{vis} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ q_x \end{bmatrix}, \quad G_{vis} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ q_y \end{bmatrix}, \quad H_{vis} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ q_z \end{bmatrix}. \quad (3.20)$$

The total enthalpy, H_{total} , is related to the total energy by

$$H_{total} = \epsilon_t + \frac{p}{\rho} \quad (3.21)$$

The viscous stress and heat flux terms may be expressed as

$$\tau_{xx} = 2\mu \left(\frac{\partial v_x}{\partial x} \right) + \lambda_v \nabla \cdot \vec{V}, \quad (3.22)$$

$$\tau_{xy} = \mu \left[\left(\frac{\partial v_y}{\partial x} \right) + \left(\frac{\partial v_x}{\partial y} \right) \right], \quad (3.23)$$

$$\tau_{xz} = \mu \left[\left(\frac{\partial v_y}{\partial z} \right) + \left(\frac{\partial v_z}{\partial x} \right) \right], \quad (3.24)$$

$$\tau_{yy} = 2\mu \left(\frac{\partial v_y}{\partial y} \right) + \lambda_v \nabla \cdot \vec{V} \quad (3.25)$$

$$\tau_{yz} = \mu \left[\left(\frac{\partial v_y}{\partial z} \right) + \left(\frac{\partial v_z}{\partial y} \right) \right], \quad (3.26)$$

$$\tau_{zz} = 2\mu \left(\frac{\partial v_z}{\partial z} \right) + \lambda_v \nabla \cdot \vec{V} \quad (3.27)$$

$$q_x = v_x \tau_{xx} + v_y \tau_{xy} + v_z \tau_{xz} + k \frac{\partial T}{\partial x} \quad (3.28)$$

$$q_y = v_x \tau_{yx} + v_y \tau_{yy} + v_z \tau_{yz} + k \frac{\partial T}{\partial y} \quad (3.29)$$

$$q_z = v_x \tau_{zx} + v_y \tau_{zy} + v_z \tau_{zz} + k \frac{\partial T}{\partial z} \quad (3.30)$$

where μ is the first coefficient of viscosity, λ_v is the second coefficient of viscosity, and

$$\nabla \cdot \vec{V} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \quad (3.31)$$

The remaining viscous stress terms are defined through the identities

$$\tau_{yx} = \tau_{xy}, \quad (3.32)$$

$$\tau_{zy} = \tau_{yz}, \quad (3.33)$$

$$\tau_{zx} = \tau_{xz}, \quad (3.34)$$

3.2.4 Governing Equations for Cylindrical Coordinate Solution

In this section, the governing equations for a rotating cylindrical coordinate system solution are developed. The rotating coordinate system permits the solution of rotating geometries such as turbomachinery blade rows. The rotation is always assumed to be about the x axis. In this discussion, since all solutions for turbulent flow employ the Boussinesq approximation, the overscores denoting time averaged (e.g. $\bar{\rho}$) and density weighted time averaged (e.g. \bar{r}_r) have been removed for simplicity.

The Reynolds-averaged Navier-Stokes equations for a rotating cylindrical coordinate system may be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_{inv}}{\partial x} + \frac{\partial G_{inv}}{\partial r} + \frac{1}{r} \frac{\partial H_{inv}}{\partial \theta} = K + \frac{\partial F_{vis}}{\partial x} + \frac{\partial G_{vis}}{\partial r} + \frac{1}{r} \frac{\partial H_{vis}}{\partial \theta} \quad (3.35)$$

For solutions employing the cylindrical coordinate system, the vector form of the equations contains only minor deviations from the Cartesian form, but the components of the solution and flux vectors must be redefined. For a cylindrical coordinate solution, the vector of dependent variables Q is defined as

$$Q = \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_r \\ \rho v_\theta \\ \rho \epsilon_t \end{bmatrix} \quad (3.36)$$

where the velocity components v_x , v_r , and v_θ are the absolute velocity components in the axial, radial, and circumferential coordinate directions, respectively.

The flux vectors are expressed as

$$F_{inv} = \begin{bmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_r \\ \rho v_x v_\theta \\ \rho v_x w_{rel} \\ \rho v_x H_{total} \end{bmatrix}, \quad G_{inv} = \begin{bmatrix} \rho v_r \\ \rho v_x v_r \\ \rho v_r^2 + p \\ \rho v_r v_\theta \\ \rho v_r w_{rel} \\ \rho v_r H_{total} \end{bmatrix}, \quad H_{inv} = \begin{bmatrix} \rho v_\theta \\ \rho v_x v_\theta \\ \rho v_r v_\theta \\ \rho v_\theta w_{rel} + p \\ \rho w_{rel} H_{total} \end{bmatrix} \quad (3.37)$$

$$F_{vis} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xr} \\ \tau_{x\theta} \\ q_x \end{bmatrix}, \quad G_{vis} = \begin{bmatrix} 0 \\ \tau_{rx} \\ \tau_{rr} \\ \tau_{r\theta} \\ q_r \end{bmatrix}, \quad H_{vis} = \begin{bmatrix} 0 \\ \tau_{\theta x} \\ \tau_{\theta r} \\ \tau_{\theta\theta} \\ q_\theta \end{bmatrix}, \quad (3.38)$$

and the cylindrical coordinate system source term becomes

$$K = \begin{bmatrix} 0 \\ 0 \\ \frac{\rho v_\theta^2 + p}{r} - \frac{\tau_{\theta\theta}}{r} \\ -\frac{\rho v_r v_\theta}{r} + \frac{\tau_{r\theta}}{r} \\ 0 \end{bmatrix} \quad (3.39)$$

The total enthalpy, H , is related to the total energy by

$$H_{total} = \epsilon_t + \frac{p}{\rho} \quad (3.40)$$

The viscous stress and heat flux terms may be expressed as

$$\tau_{xx} = 2\mu \left(\frac{\partial v_x}{\partial x} \right) + \lambda_v \nabla \cdot \vec{V}. \quad (3.41)$$

$$\tau_{xr} = \mu \left[\left(\frac{\partial v_r}{\partial x} \right) + \left(\frac{\partial v_x}{\partial r} \right) \right], \quad (3.42)$$

$$\tau_{x\theta} = 2\mu \left[\left(\frac{\partial v_\theta}{\partial x} \right) + \left(\frac{1}{r} \frac{\partial v_x}{\partial \theta} \right) \right], \quad (3.43)$$

$$\tau_{rr} = 2\mu \left(\frac{\partial v_r}{\partial r} \right) + \lambda_v \nabla \cdot \vec{V} \quad (3.44)$$

$$\tau_{r\theta} = \mu \left[\left(\frac{1}{r} \frac{\partial v_r}{\partial \theta} \right) + \left(r \frac{\partial v_\theta}{\partial r} \right) \right], \quad (3.45)$$

$$\tau_{\theta\theta} = 2\mu \left(\frac{1}{r} \frac{\partial v_\theta}{\partial \theta} \right) + \lambda_v \nabla \cdot \vec{V} + 2\mu \frac{v_r}{r} \quad (3.46)$$

$$q_x = v_x \tau_{xx} + v_r \tau_{xr} + v_\theta \tau_{x\theta} + k \frac{\partial T}{\partial x} \quad (3.47)$$

$$q_r = v_x \tau_{rx} + v_r \tau_{rr} + v_\theta \tau_{r\theta} + k \frac{\partial T}{\partial r} \quad (3.48)$$

$$q_\theta = v_x \tau_{\theta x} + v_r \tau_{\theta r} + v_\theta \tau_{\theta\theta} + k \frac{\partial T}{\partial \theta} \quad (3.49)$$

where μ is the first coefficient of viscosity, λ_v is the second coefficient of viscosity, and

$$\nabla \cdot \vec{V} = \frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} \quad (3.50)$$

The remaining viscous stress terms are defined through the identities

$$\tau_{rx} = \tau_{xr}, \quad (3.51)$$

$$\tau_{\theta r} = \tau_{r\theta}, \quad (3.52)$$

$$\tau_{\theta x} = \tau_{x\theta}, \quad (3.53)$$

3.3 Fluid Properties

The primary working fluid is assumed to be air acting as a perfect gas, thus the ideal gas equation of state has been used. Fluid properties such as specific heats, specific heat ratio, and Prandtl number are assumed to be constant. The fluid viscosity is temperature dependent and is derived from the Sutherland (see e.g. [42]) formula:

$$\mu = C_1 \frac{(T)^{\frac{3}{2}}}{T + C_2} \quad (3.54)$$

where for air the coefficients are specified as:

$$C_1 = 2.27 \cdot 10^{-8} \frac{lbm}{ft \cdot sec} \quad C_2 = 198.72 degrees R$$

The so-called second coefficient of viscosity λ_v is fixed according to:

$$\lambda_v = -\frac{2}{3}\mu \quad (3.55)$$

The thermal conductivity is determined from the viscosity and the definition of the Prandtl number as:

$$k = \frac{c_p \mu}{Pr} \quad (3.56)$$

3.4 Numerical Formulation

The numerical formulation for the *ADPAC07* code is provided in the subsections below.

3.4.1 Finite Volume Discretization

Integration of the three-dimensional differential form of the Navier-Stokes equations over a finite control volume yields an equation of the form:

$$\int \int \int \frac{\partial}{\partial t}(Q)dV + L_{inv}(Q) = L_{vis}(Q) + \int \int \int DdV \quad (3.57)$$

where:

$$L_{inv}(Q) = \int \int_{dA} [F_{inv}dA_1 + G_{inv}dA_2 + H_{inv}dA_3] \quad (3.58)$$

and:

$$L_{vis}(Q) = \int \int_{dA} [F_{vis}dA_1 + G_{vis}dA_2 + H_{vis}dA_3] \quad (3.59)$$

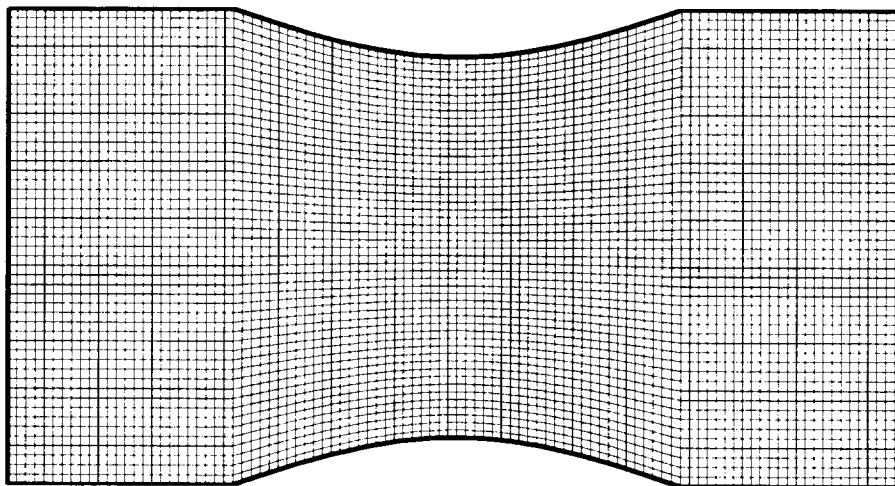
The Gauss divergence theorem has been employed to convert several volume integrals to surface flux integrals, which simplifies the numerical evaluation of many terms (see e.g. [42]). The inviscid (convective) and viscous (diffusive) flux contributions are expressed separately by the operators L_{inv} and L_{vis} , respectively. The vector of dependent variables Q and other terms are described separately for both a Cartesian and a cylindrical coordinate system above.

The discrete numerical solution is developed from the integral governing equations derived in the previous sections by employing a finite volume solution procedure. This procedure closely follows the basic scheme described by Jameson [31]. In order to appreciate and utilize the features of the *ADPAC07* solution system, the concept of a multiple block grid system must be fully understood. It is expected that the reader possesses at least some understanding of the concepts of computational fluid dynamics (CFD), so the use of a numerical grid to discretize a flow domain should not be foreign. Many CFD analyses rely on a single structured ordering of grid points upon which the numerical solution is performed. Multiple block grid systems are different only in that several structured grid systems are used in harmony to generate the numerical solution. The domain of interest is subdivided into one or more structured arrays of hexahedral cells. Each array of cells is referred to as a “block”, and the overall scheme is referred to as a multiple blocked mesh solver as a result of the ability to manage more than one block. This concept is illustrated graphically in two dimensions for the flow through a nozzle in Figures 3.2-3.4.

The grid system in Figure 3.2 employs a single structured ordering, resulting in a single computational space to contend with. The mesh system in Figure 3.3 is comprised of two,

ADPAC 2-D Nozzle Single Block Mesh Structure Illustration

Physical Domain



Computational Domain

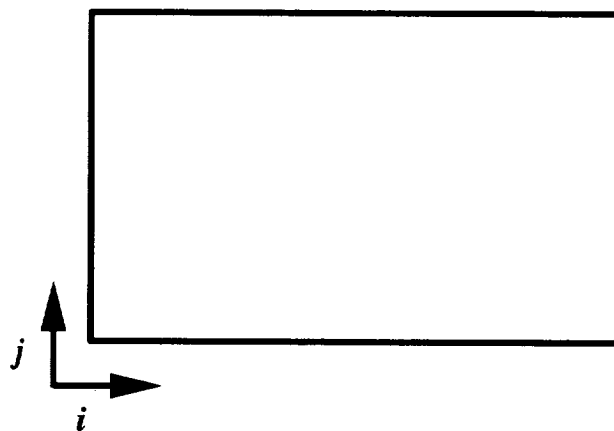


Figure 3.2: ADPAC07 2-D single block mesh structure illustration.

ADPAC 2-D Nozzle Two Block Mesh Structure Illustration

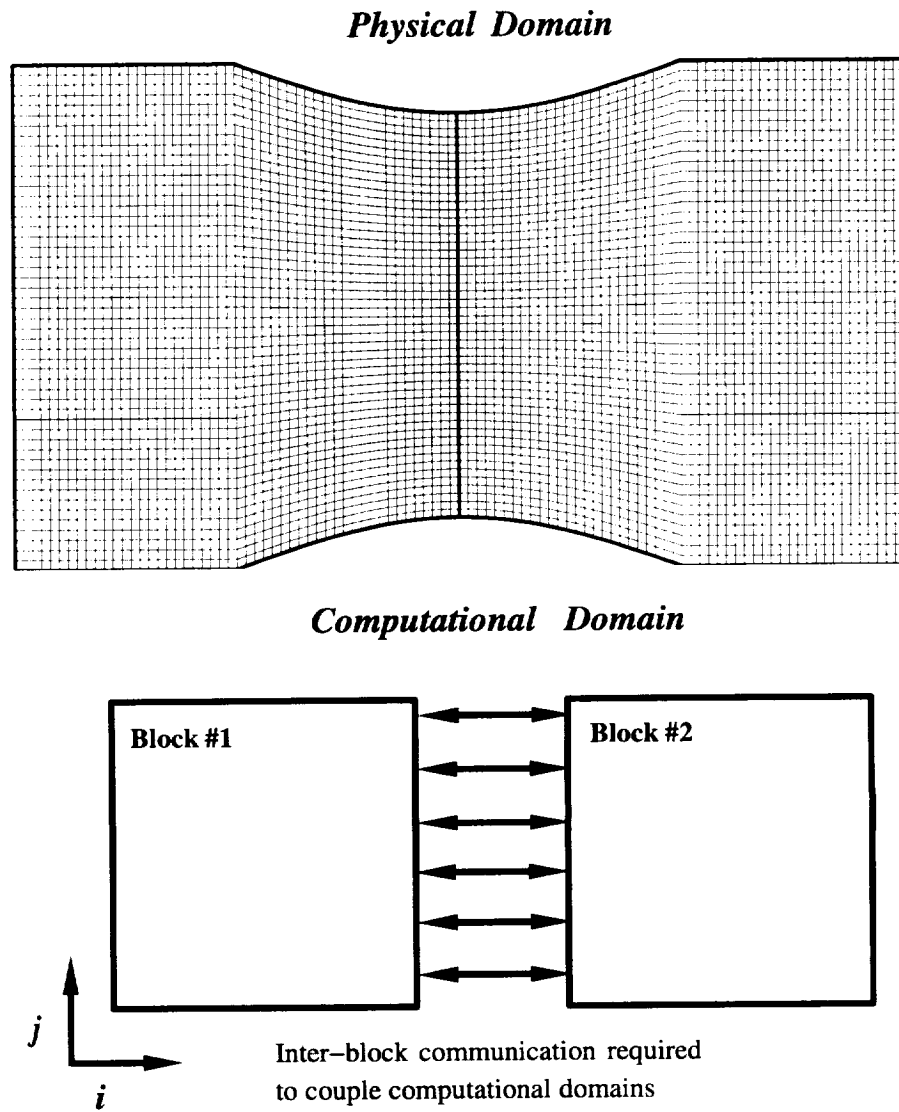
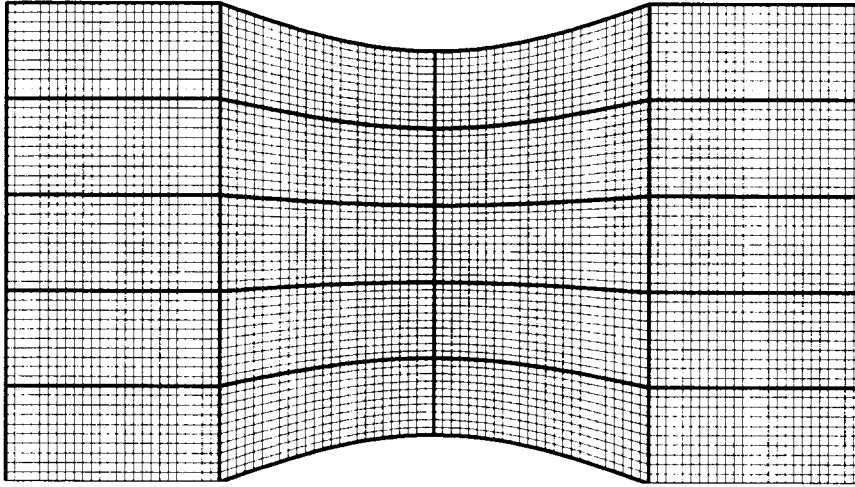


Figure 3.3: *ADPAC07* 2-D two block mesh structure illustration.

ADPAC 2-D Nozzle Multiple Block Mesh Structure Illustration
Physical Domain



Computational Domain

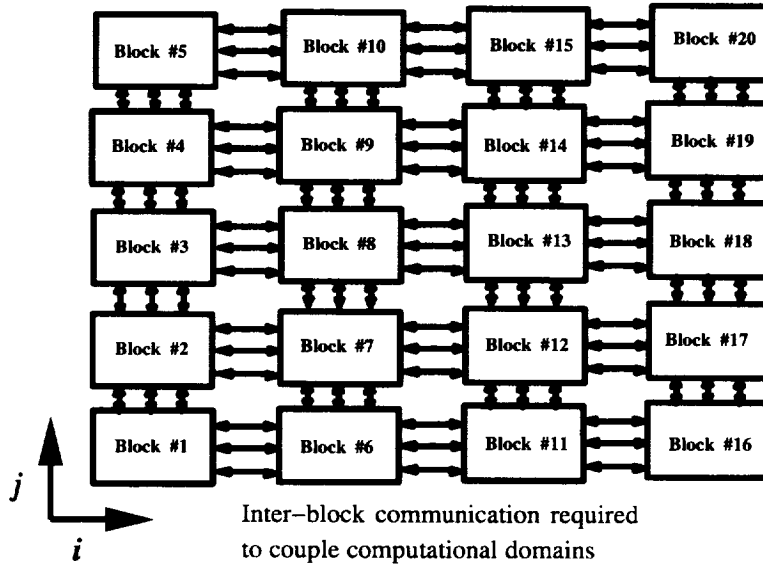


Figure 3.4: ADPAC07 2-D multiple block mesh structure illustration.

separate structured grid blocks, and consequently, the numerical solution consists of two unique computational domains. In theory, the nozzle flowpath could be subdivided into any number of domains employing structured grid blocks resulting in an identical number of computational domains to contend with, as shown in the 20 block decomposition illustrated in Figure 3.4. The complicating factor in this domain decomposition approach is that the numerical solution must provide a means for the isolated computational domains to communicate with each other in order to satisfy the conservation laws governing the desired aerodynamic solution. Hence, as the number of subdomains used to complete the aerodynamic solution grows larger, the number of inter-domain communication paths increases in a corresponding manner. (It should be noted that this domain decomposition/communication overhead relationship is also a key concept in parallel processing for large scale computations. The natural parallelization afforded by the multiple block mesh domain decomposition is the fundamental basis for the *ADPAC07* code parallelization described later in this report.) Clearly, it is often not possible to generate a single structured grid to encompass the domain of interest without sacrificing grid quality, and therefore, a multiple block grid system has significant advantages.

The *ADPAC07* code was developed to utilize the multiple block grid concept to full extent by permitting an arbitrary number of structured grid blocks with user specifiable communication paths between blocks. The inter-block communication paths are implemented as a series of boundary conditions on each block which, in some cases, communicate flow information from one block to another. The advantages of the multiple block solution concept are exploited in the calculations presented in later chapters as a means of treating complicated geometries with multiple blade rows of varying blade number, and to exploit computational enhancements such as multigrid.

The solution for each mesh block in a multiple block grid is computed identically, and therefore the numerical approach is described for a single mesh block. In any given mesh block, the numerical grid is used to define a set of hexahedral cells, the vertices of which are defined by the eight surrounding mesh points. This construction is illustrated in Figure 3.5.

The cell face surface area normal vector components dA_x , dA_y , and dA_z are calculated using the cross product of the diagonals defined by the four vertices of the given face, and the cell volume is determined by a procedure outlined by Hung and Kordulla [43] for generalized nonorthogonal cells. The integral relations expressed by the governing equations are determined for each cell by approximating the area-integrated convective and diffusive fluxes with a representative value along each cell face, and by approximating the volume-integrated terms with a representative cell volume weighted value. The discrete numerical approximation to the governing equation then becomes

$$\begin{aligned}
 (\mathcal{V}) \frac{Q_{i,j,k}^{n+1} - Q_{i,j,k}^n}{\Delta t} = & (F_{inv}(Q)_{i+\frac{1}{2},j,k} - F_{inv}(Q)_{i-\frac{1}{2},j,k} \\
 & + G_{inv}(Q)_{i,j+\frac{1}{2},k} - G_{inv}(Q)_{i,j-\frac{1}{2},k} \\
 & + H_{inv}(Q)_{i,j,k+\frac{1}{2}} - H_{inv}(Q)_{i,j,k-\frac{1}{2}} \\
 & + F_{vis}(Q)_{i+\frac{1}{2},j,k} - F_{vis}(Q)_{i-\frac{1}{2},j,k} \\
 & + G_{vis}(Q)_{i,j+\frac{1}{2},k} - G_{vis}(Q)_{i,j-\frac{1}{2},k} \\
 & + H_{vis}(Q)_{i,j,k+\frac{1}{2}} - H_{vis}(Q)_{i,j,k-\frac{1}{2}}) \\
 & + (\mathcal{V})K + D_{i,j,k}(Q)
 \end{aligned} \tag{3.60}$$

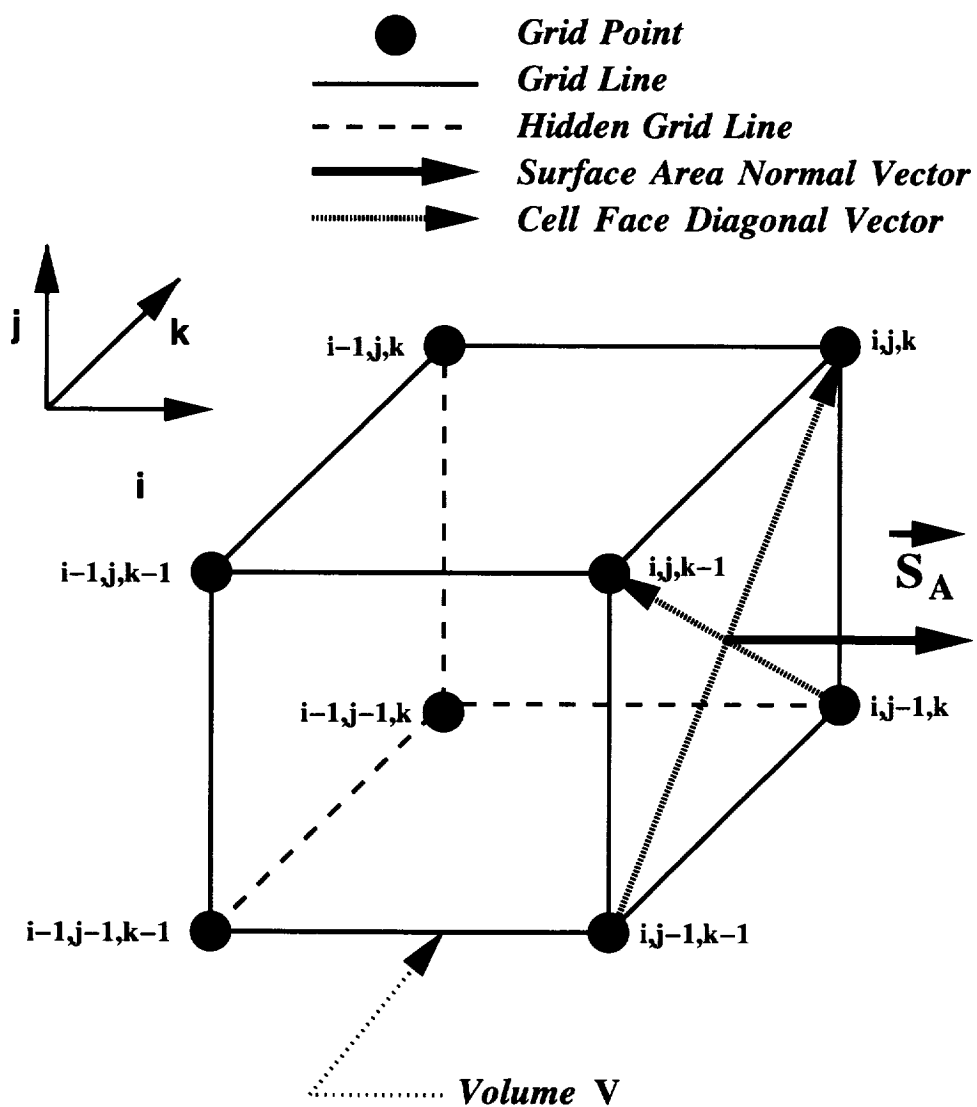


Figure 3.5: Three-dimensional finite volume cell.

Following the algorithm defined by Jameson [31], it is convenient to store the flow variables as a representative value for the interior of each cell, and thus the scheme is referred to as cell-centered. Here, i, j, k represents the local cell indices in the structured cell-centered array, \mathcal{V} is the local cell volume, Δt is the calculation time interval, and $D_{i,j,k}$ is an artificial numerical dissipation function which is added to the governing equations to aid numerical stability, and to eliminate spurious numerical oscillations in the vicinity of flow discontinuities such as shock waves. Indicial expressions such as $i + \frac{1}{2}, j, k$ represents data evaluated at the cell face, or interface between two adjacent volumes. The discrete convective fluxes are constructed by using a representative value of the flow variables Q which is determined by an algebraic average of the values of Q in the cells lying on either side of the local cell face. A conceptual illustration of the finite-volume, cell centered data approach, and the subsequent convective flux evaluation process for a cell face are given on Figure 3.6. Viscous stress terms and thermal conduction terms are constructed by applying a generalized coordinate transformation to the governing equations as follows:

$$\xi = \xi(x, y, z), \quad \eta = \eta(x, y, z), \quad \zeta = \zeta(x, y, z) \quad (3.61)$$

The chain rule may then be used to expand the various derivatives in the viscous stresses as:

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial}{\partial \zeta}, \quad (3.62)$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial y} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial}{\partial \zeta}, \quad (3.63)$$

$$\frac{\partial}{\partial z} = \frac{\partial \xi}{\partial z} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial}{\partial \zeta}. \quad (3.64)$$

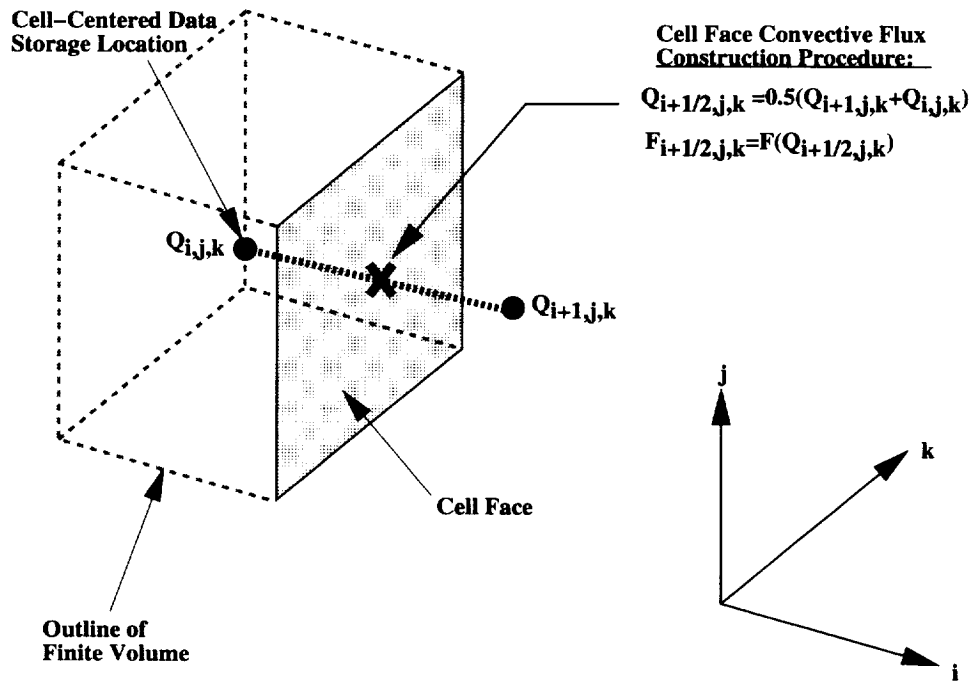
The transformed derivatives may now be easily calculated by differencing the variables in computational space (i corresponds to the ξ direction, j corresponds to the η direction, and k corresponds to the ζ direction), and utilizing the appropriate identities for the metric differences (see e.g. [42]). This process is illustrated schematically in Figure 3.7.

3.4.2 Runge-Kutta Time Integration

The time-stepping scheme used to advance the discrete numerical representation of the governing equations is a multistage Runge-Kutta integration. An m stage Runge-Kutta integration for the discretized equations is expressed as:

$$\begin{aligned} Q_1 &= Q^n - \alpha_1 \Delta t [L(Q^n) + D(Q^n)], \\ Q_2 &= Q^n - \alpha_2 \Delta t [L(Q_1) + D(Q^n)], \\ Q_3 &= Q^n - \alpha_3 \Delta t [L(Q_2) + D(Q^n)], \\ Q_4 &= Q^n - \alpha_4 \Delta t [L(Q_3) + D(Q^n)], \\ &\dots \\ &\dots \\ Q_m &= Q^n - \alpha_m \Delta t [L(Q_{m-1}) + D(Q^n)], \\ Q^{n+1} &= Q_m \end{aligned} \quad (3.65)$$

ADPAC Cell Face Convective Flux Evaluation



ADPAC Cell Face Area Normal Vector Evaluation

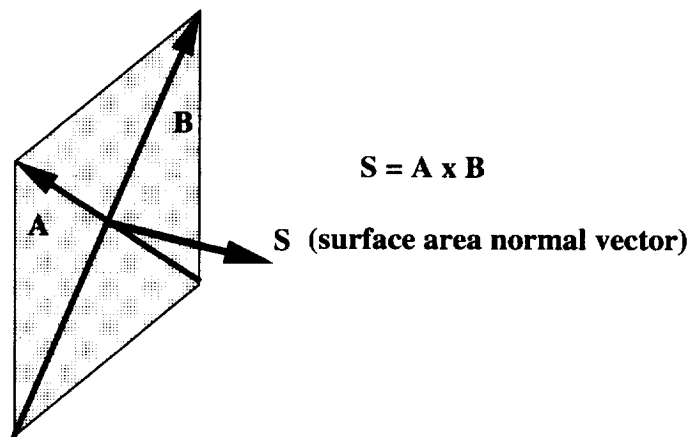


Figure 3.6: ADPAC07 finite volume cell centered data configuration and convective flux evaluation process.

ADPAC Cell Face Diffusive Flux Evaluation

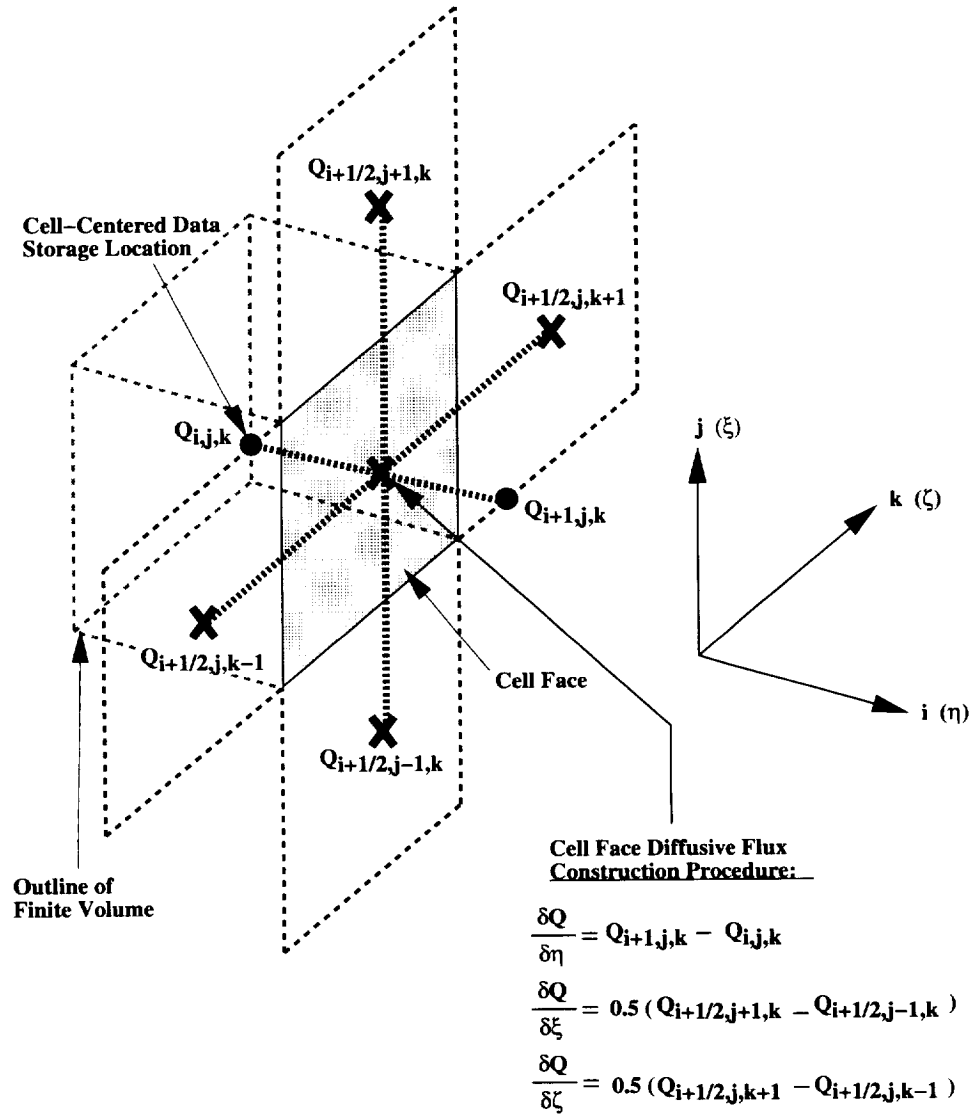


Figure 3.7: ADPAC'07 finite volume cell centered data configuration and diffusive flux evaluation process.

where:

$$L(Q) = L_{inv}(Q) - L_{vis}(Q) \quad (3.66)$$

For simplicity, viscous flux contributions to the discretized equations are only calculated for the first stage, and the values are frozen for the remaining stages. This reduces the overall computational effort and does not appear to significantly alter the solution. It is also generally not necessary to recompute the added numerical dissipation terms during each stage. Three different multistage Runge-Kutta schemes (2 four-stage schemes, and 1 five-stage scheme) are available in the *ADPAC07* code, but only the four-stage time-marching scheme described below was utilized for the calculations presented in this report.

The coefficients for the four stage Runge-Kutta time-marching scheme employed in this study are listed below:

$$\alpha_1 = \frac{1}{8}, \quad \alpha_2 = \frac{1}{4}, \quad \alpha_3 = \frac{1}{2}, \quad \alpha_4 = 1 \quad (3.67)$$

A linear stability analysis of the four stage Runge-Kutta time-stepping scheme utilized during this study indicate that the scheme is stable for all calculation time increments δt which satisfy the stability criteria $CFL \leq 2\sqrt{2}$. Based on convection constraints alone, the *CFL* number may be defined in a one-dimensional manner as:

$$CFL = \frac{\Delta t}{\frac{|v_x| + a}{\Delta x}} \quad (3.68)$$

In practice, the calculation time interval must also include restrictions resulting from diffusion phenomena. The time step used in the numerical calculation results from both convective and diffusive considerations and is calculated as:

$$\Delta t = CFL \left(\frac{1.0}{\lambda_i + \lambda_j + \lambda_k + \nu_i + \nu_j + \nu_k} \right) \quad (3.69)$$

where the convective and diffusive coordinate wave speeds (λ and ν , respectively) are defined as:

$$\lambda_i = \mathcal{V}/(\vec{V} \cdot \vec{S}_i + a) \quad (3.70)$$

$$\nu_i = \frac{\rho(\mathcal{V})^2}{C_{\Delta t}(S^2)\mu} \quad (3.71)$$

The factor $C_{\Delta t}$ is a “safety factor” of sorts, which must be imposed as a result of the limitations of the linear stability constraints for a set of equations which are truly nonlinear. This factor was determined through numerical experimentation and normally ranges from 2.5-7.5.

For steady flow calculations, an acceleration technique known as local time stepping is used to enhance convergence to the steady-state solution. Local time stepping utilizes the maximum allowable time increment at each point during the course of the solution. While this destroys the physical nature of the transient solution, the steady-state solution is unaffected and can be obtained in fewer iterations of the time-stepping scheme. For unsteady flow calculations, of course, a uniform value of the time step Δt must be used at every grid point to maintain the time-accuracy of the solution. Other convergence enhancements such as implicit residual smoothing and multigrid (described in later sections) are also applied for steady flow calculations.

3.4.3 Dissipation Function

In order to prevent odd-even decoupling of the numerical solution, nonphysical oscillations near shock waves, and to obtain rapid convergence for steady state solutions, artificial dissipative terms are added to the discrete numerical representation of the governing equations. The added dissipation model is based on the combined works of Jameson et al. [31], Martinelli [50], and Swanson et al. [44]. A blend of fourth and second differences is used to provide a third order background dissipation in smooth flow regions and first order dissipation near discontinuities. The discrete equation dissipative function is given by:

$$D_{i,j,k}(Q) = (D_i^2 - D_i^4 + D_j^2 - D_j^4 + D_k^2 - D_k^4)Q_{i,j,k} \quad (3.72)$$

The second and fourth order dissipation operators are determined by

$$D_\xi^2 Q_{i,j,k} = \nabla_\xi((\lambda_\xi)_{i+\frac{1}{2}} \epsilon_{i+\frac{1}{2},j,k}^2) \Delta_\xi Q_{i,j,k} \quad (3.73)$$

$$D_\xi^4 Q_{i,j,k} = \nabla_\xi((\lambda_\xi)_{i+\frac{1}{2}} \epsilon_{i+\frac{1}{2},j,k}^4) \Delta_\xi \nabla_\xi \Delta_\xi Q_{i,j,k} \quad (3.74)$$

where Δ_ξ and ∇_ξ are forward and backward difference operators in the ξ direction. In order to avoid excessively large levels of dissipation for cells with large aspect ratios, and to maintain the damping properties of the scheme, a variable scaling of the dissipative terms is employed which is an extension of the two dimensional scheme given by Martinelli [50]. The scaling factor is defined as a function of the spectral radius of the Jacobian matrices associated with the ξ , η , and ζ directions and provides a scaling mechanism for varying cell aspect ratios through the following scheme:

$$(\lambda_\xi)_{i+\frac{1}{2},j,k} = (\lambda_\xi)_{i+\frac{1}{2},j,k} \Phi_{i+\frac{1}{2},j,k} \quad (3.75)$$

The function Φ controls the relative importance of dissipation in the three coordinate directions as:

$$\Phi_{i+\frac{1}{2},j,k} = 1 + \max \left(\left(\frac{(\lambda_\eta)_{i+\frac{1}{2},j,k}}{(\lambda_\xi)_{i+\frac{1}{2},j,k}} \right)^\alpha, \left(\frac{(\lambda_\zeta)_{i+\frac{1}{2},j,k}}{(\lambda_\xi)_{i+\frac{1}{2},j,k}} \right)^\alpha \right) \quad (3.76)$$

The directional eigenvalue scaling functions are defined by:

$$(\lambda_\xi)_{i+\frac{1}{2},j,k} = U_{i+\frac{1}{2},j,k} (S_\xi)_{i+\frac{1}{2},j,k} + c(S_\xi)_{i+\frac{1}{2},j,k} \quad (3.77)$$

$$(\lambda_\eta)_{i+\frac{1}{2},j,k} = U_{i+\frac{1}{2},j,k} (S_\eta)_{i+\frac{1}{2},j,k} + c(S_\eta)_{i+\frac{1}{2},j,k} \quad (3.78)$$

$$(\lambda_\zeta)_{i+\frac{1}{2},j,k} = U_{i+\frac{1}{2},j,k} (S_\zeta)_{i+\frac{1}{2},j,k} + c(S_\zeta)_{i+\frac{1}{2},j,k} \quad (3.79)$$

The use of the maximum function in the definition of Φ is important for grids where λ_η/λ_ξ and $\lambda_\zeta/\lambda_\xi$ are very large and of the same order of magnitude. In this case, if these ratios are summed rather than taking the maximum, the dissipation can become too large, resulting in degraded solution accuracy and poor convergence. Because three-dimensional solution grids tend to exhibit large variations in the cell aspect ratio, there is less freedom in the choice of the parameter α for this scheme, and a value of 0.5 was found to provide a robust scheme.

The coefficients in the dissipation operator use the solution pressure as a sensor for the presence of shock waves in the solution and are defined as:

$$\epsilon_{i+\frac{1}{2},j,k}^2 = \kappa^2 \max(\nu_{i-1,j,k}, \nu_{i,j,k}, \nu_{i+1,j,k}, \nu_{i+2,j,k}) \quad (3.80)$$

$$\nu_{i,j,k} = \frac{|(p_{i-1,j,k} - 2p_{i,j,k} + p_{i+1,j,k})|}{(p_{i-1,j,k} + 2p_{i,j,k} + p_{i+1,j,k})} \quad (3.81)$$

$$\epsilon_{i+\frac{1}{2},j,k}^4 = \max(0, \kappa^4 - \epsilon_{i+\frac{1}{2},j,k}^2) \quad (3.82)$$

where κ^2, κ^4 are user-defined constants. Typical values for these constants are

$$\kappa^2 = \frac{1}{2} \quad \kappa^4 = \frac{1}{64} \quad (3.83)$$

The dissipation operators in the η and ζ directions are defined in a similar manner.

3.4.4 Implicit Residual Smoothing

The stability range of the basic time-stepping scheme can be extended using implicit smoothing of the residuals. This technique was described by Hollanders et al. [51] for the Lax-Wendroff scheme and later developed by Jameson [31] for the Runge-Kutta scheme. Since an unsteady flow calculation for a given geometry and grid is likely to be computationally more expensive than a similar steady flow calculation, it would be advantageous to utilize this acceleration technique for time-dependent flow calculations as well. In an analysis of two dimensional unsteady flows, Jorgensen and Chima [33] demonstrated that a variant of the implicit residual smoothing technique could be incorporated into a time-accurate explicit method to permit the use of larger calculation time increments without adversely affecting the results of the unsteady calculation. The implementation of this residual smoothing scheme reduced the CPU time for their calculation by a factor of five. This so-called time-accurate implicit residual smoothing operator was then also demonstrated by Rao and Delaney [52] for a similar two-dimensional unsteady calculation and by Hall, et al. [23],[24] for several three-dimensional time-dependent flows. Although this “time-accurate” implicit residual smoothing scheme is not developed theoretically to accurately provide the unsteady solution, it can be demonstrated that errors introduced through this residual smoothing process are very local in nature, and are generally not greater than the discretization error.

The standard implicit residual smoothing operator can be written as:

$$(1 - \epsilon \Delta \nabla) R_m^* = R_m \quad (3.84)$$

To simplify the numerical implementation, this standard operator is traditionally approximately factored into the following coordinate specific form:

$$(1 - \epsilon_\xi \Delta_\xi \nabla_\xi)(1 - \epsilon_\eta \Delta_\eta \nabla_\eta)(1 - \epsilon_\zeta \Delta_\zeta \nabla_\zeta) R_m = R_m \quad (3.85)$$

where the residual R_m is defined as:

$$R_m = \alpha_m \frac{\Delta t}{V} (Q_m - D_m), \quad m = 1, mstages \quad (3.86)$$

for each of the m stages in the Runge-Kutta multistage scheme. Here Q_m is the sum of the convective and diffusive terms, D_m the total dissipation at stage m , and R_m the final (smoothed) residual at stage m .

The smoothing reduction is applied sequentially in each coordinate direction as:

$$R_m^* = (1 - \epsilon_\xi \Delta_\xi \nabla_\xi)^{-1} R_m$$

$$\begin{aligned}
R_m^{**} &= (1 - \epsilon_\eta \Delta_\eta \nabla_\eta)^{-1} R_m^* \\
R_m^{***} &= (1 - \epsilon_\zeta \Delta_\zeta \nabla_\zeta)^{-1} R_m^{**} \\
\bar{R}_m &= R_m^{***}
\end{aligned} \tag{3.87}$$

where each of the first three steps above requires the inversion of a scalar tridiagonal matrix. In general, it is desirable to apply the smoothing at each stage of the Runge-Kutta time-marching procedure.

The use of constant coefficients (ϵ) in the implicit treatment has proven to be useful, even for meshes with high aspect ratio cells, provided additional support such as enthalpy damping (see [31]) is introduced. Unfortunately, the use of enthalpy damping, which assumes a constant total enthalpy throughout the flowfield, cannot be used for an unsteady flow, and many steady flows where the total enthalpy may vary. It has been shown that the need for enthalpy damping can be eliminated by using variable coefficients in the implicit treatment which account for the variation of the cell aspect ratio. Martinelli [50] derived a functional form for the variable coefficients for two-dimensional flows which are functions of characteristic wave speeds. In this study, the three-dimensional extension described by Radespiel et al. [44] is utilized, and is expressed as:

$$\epsilon_\xi = \max \left(0, \frac{1}{4} \left[\frac{CFL}{CFL_{max}} \frac{1 + \max(r_{\eta\xi}^\alpha r_{\zeta\xi}^\alpha)}{1 + \max(r_{\eta\xi} r_{\zeta\xi})} \right]^2 - 1 \right) \tag{3.88}$$

$$\epsilon_\eta = \max \left(0, \frac{1}{4} \left[\frac{CFL}{CFL_{max}} \frac{1 + \max(r_{\xi\eta}^\alpha r_{\zeta\eta}^\alpha)}{1 + \max(r_{\xi\eta} r_{\zeta\eta})} \right]^2 - 1 \right) \tag{3.89}$$

$$\epsilon_\zeta = \max \left(0, \frac{1}{4} \left[\frac{CFL}{CFL_{max}} \frac{1 + \max(r_{\eta\zeta}^\alpha r_{\xi\zeta}^\alpha)}{1 + \max(r_{\eta\zeta} r_{\xi\zeta})} \right]^2 - 1 \right) \tag{3.90}$$

CFL represents the local value of the CFL number based on the calculation time increment Δt , and CFL_{max} represents the maximum stable value of the CFL number permitted by the unmodified scheme (normally, in practice, this is chosen as 2.5 for a four stage scheme and 3.5 for a five stage scheme, although linear stability analysis suggests that $2\sqrt{2}$, and 3.75 are the theoretical limits for the four and five stage schemes, respectively). From this formulation it is obvious then that the residual smoothing operator is only applied in those regions where the local CFL number exceeds the stability-limited value. In this approach, the residual operator coefficient becomes zero at points where the local CFL number is less than that required by stability, and the influence of the smoothing is only locally applied to those regions exceeding the stability limit. Practical experience involving unsteady flow calculations suggests that for a constant time increment, the majority of the flowfield utilizes CFL numbers less than the stability-limited value to maintain a reasonable level of accuracy. Local smoothing is therefore typically required only in regions of small grid spacing, where the stability-limited time step is very small. Numerical tests both with and without the time-accurate implicit residual smoothing operator for the flows of interest in this study were found to produce essentially identical results, while the time-accurate residual smoothing resulted in a decrease in CPU time by a factor of 2-3. In practice, the actual limit on the calculation CFL number were determined to be roughly twice the values specified for CFL_{max} , above.

Multigrid Mesh Level Decomposition

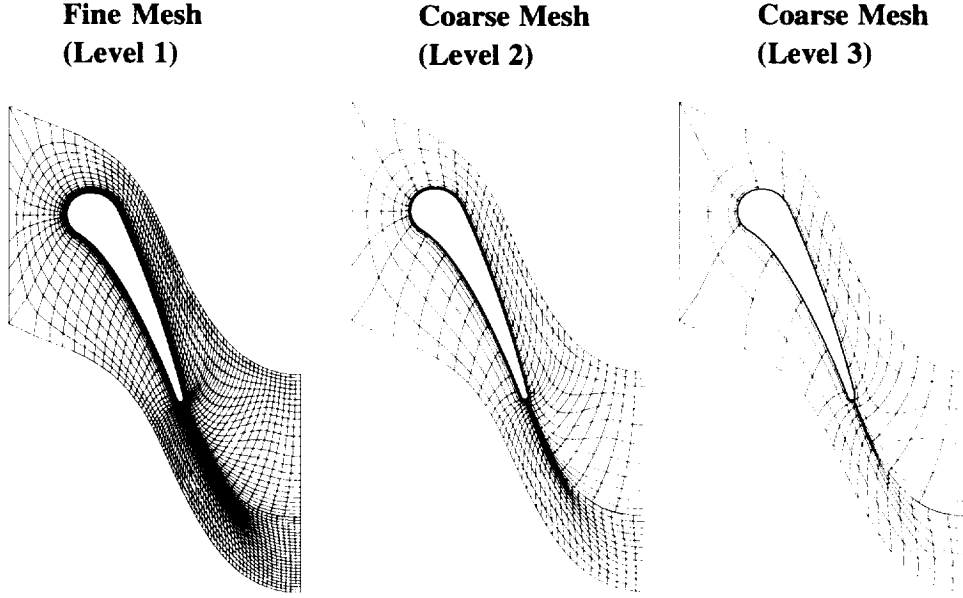


Figure 3.8: Multigrid mesh coarsening strategy and mesh index relation.

3.4.5 Multigrid Convergence Acceleration

Multigrid (not to be confused with a multiple blocked grid!) is a numerical solution technique which attempts to accelerate the convergence of an iterative process (such as a steady flow prediction using a time-marching scheme) by computing corrections to the solution on coarser meshes and propagating these changes to the fine mesh through interpolation. This operation may be recursively applied to several coarsenings of the original mesh to effectively enhance the overall convergence. In the present multigrid application, coarse meshes are derived from the preceding finer mesh by eliminating every other mesh line in each coordinate direction as shown in Figure 3.8. As a result, the number of multigrid levels (coarse mesh divisions) is controlled by the mesh size, and, in the case of the *ADPAC07* code, also by the indices of the embedded mesh boundaries (such as blade leading and trailing edges, etc.) (see Figure 3.8). These restrictions suggest that mesh blocks should be constructed such that the internal boundaries and overall size coincide with numbers which are compatible with the multigrid solution procedure (i.e., the mesh size should be 1 greater than any number which can be divided by 2 several times and remain whole numbers: e.g. 9, 17, 33, 65 etc.)

The multigrid procedure is applied in a V-cycle as shown in Figure 3.9, whereby the fine mesh solution is initially “injected” into the next coarser mesh, the appropriate forcing functions are then calculated based on the differences between the calculated coarse mesh residual and the residual which results from a summation of the fine mesh residuals for the coarse mesh cell, and the solution is advanced on the coarse mesh. This sequence is repeated on each successively coarser mesh until the coarsest mesh is reached. At this point, the correction to the solution ($Q_{i,j,k}^{n+1} - Q_{i,j,k}^n$) is interpolated to the next finer mesh, a new solution is defined on that mesh, and the interpolation of corrections is applied sequentially

Multigrid V-Cycle Strategy

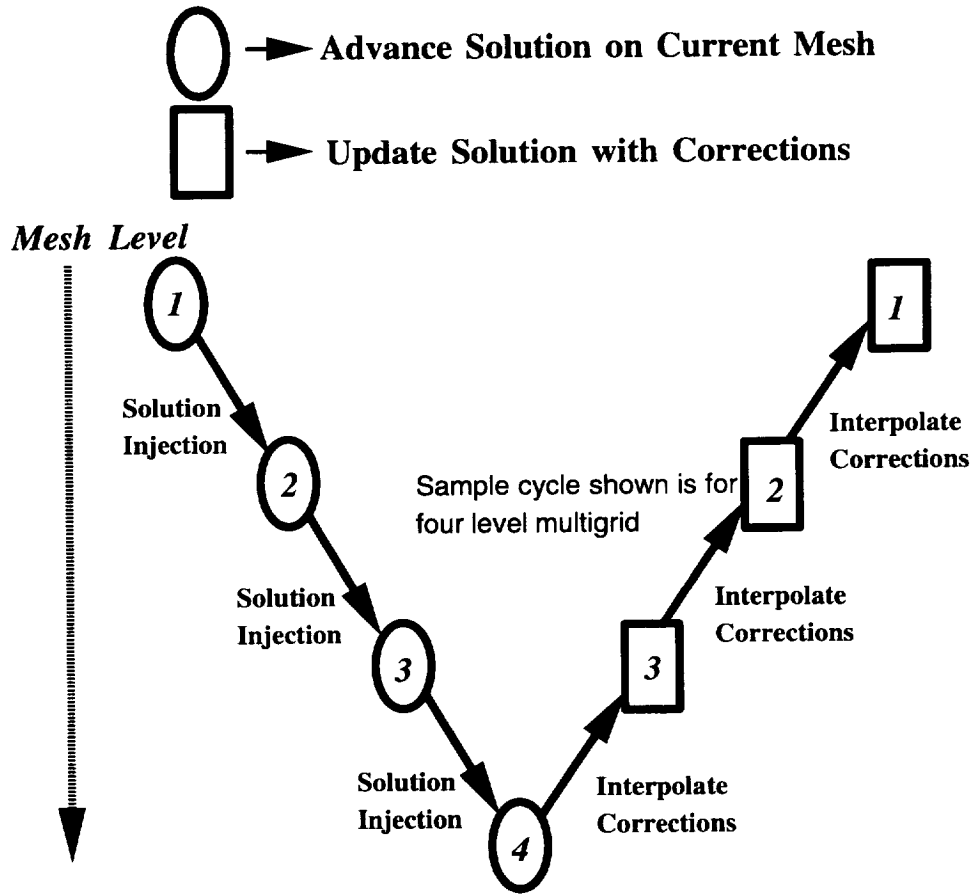


Figure 3.9: Multigrid V cycle strategy.

until the finest mesh is reached. Following a concept suggested by Swanson et al. [44], it is sometimes desirable to smooth the final corrections on the finest mesh to reduce the effects of oscillations induced by the interpolation process. A constant coefficient implementation of the implicit residual smoothing scheme described in Section 3.5 is used for this purpose. The value of the smoothing constant is normally taken to be 0.2.

A second multigrid concept which should be discussed is the so-called “full” multigrid startup procedure. The “full” multigrid method is used to initialize a solution by first computing the flow on a coarse mesh, performing several time-marching iterations on that mesh (which, by the way could be multigrid iterations if successively coarser meshes are available), and then interpolating the solution at that point to the next finer mesh, and repeating the entire process until the finest mesh level is reached. The intent here is to generate a reasonably approximate solution on the coarser meshes before undergoing the expense of the fine mesh multigrid cycles. Again, the “full” multigrid technique only applies to starting up a solution.

3.4.6 Implicit Time-Marching Algorithm Procedure

The development of an implicit time-marching strategy for the *ADPAC07* code was initiated during this study. This effort was directed at improving the computational efficiency of the *ADPAC07* code for lengthy time-dependent calculations, particularly for viscous flows, where the restricted time step of the explicit time-marching algorithm due to highly clustered meshes is prohibitively expensive. The implicit algorithm type selected was chosen to take advantage of the multigrid solution capabilities of the explicit *ADPAC07* time marching algorithm, and the various steady state convergence acceleration techniques (implicit residual smoothing, local time stepping, etc.) which have been incorporated into the code.

The implicit algorithm is best explained through derivation. The original explicit steady state iterative numerical algorithm in the *ADPAC07* code ultimately solves an equation of the form:

$$\frac{\partial Q}{\partial t} = -R(Q) \quad (3.91)$$

where Q is the vector of dependent variables, t is time, and $R(Q)$ is the residual which includes convective, diffusive, and artificial dissipation fluxes. The solution is typically advanced in time t until the residual approaches zero, or simply $\partial Q/\partial t = 0$ which implies a time-independent (steady state) solution. The modified solution scheme introduces a fictitious time τ , and solves an equation of the form:

$$\frac{\partial Q}{\partial \tau} = \frac{\partial Q}{\partial t} + R(Q) = R^*(Q) \quad (3.92)$$

Now the solution for driving the new residual $R^*(Q)$ to zero can be advanced by marching in τ using all of the previously developed steady state convergence acceleration techniques, and the final solution satisfies the equation

$$\frac{\partial Q}{\partial t} + R(Q) = 0 \quad (3.93)$$

which is the desired time-dependent solution. This approach follows the work of Jame-son [31] and the more recent applications of Arnone et al. [73]. Derivatives with respect to the real time t are discretized using either a 2 point or a 3 point backward formula which results in an implicit scheme which is second order accurate in time. The discretized equation solved at each time level therefore becomes:

$$\frac{\partial Q}{\partial \tau} = \frac{3Q^{n+1} - 4Q^n + Q^{n-1}}{2\Delta t} + R(Q^{n+1}) = R^*(Q^{n+1}) \quad (3.94)$$

where the subscript n is associated with real time. Between each real time step, then, the solution is advanced multiple iterations in the nonphysical time to satisfy the time-accurate equations. The pseudo time numerical approach was recently demonstrated for time-dependent flows in turbomachinery geometries by Arnone et al. [74].

The time discretization described above is fully implicit; however, when solved by marching in τ , stability problems can occur when the time increment in the pseudo time variable $\Delta\tau$ exceeds the physical time step Δt . Linear stability analysis (see e.g. [73]) indicates that the pseudo time increment $\Delta\tau$ must be less than $2/3CFL^*\Delta t$ where CFL^* is the ratio of the local CFL number to the maximum CFL number of the explicit time-marching scheme.

Numerical experiments for the algorithm with the physical time derivative term treated in an explicit manner indicated that the algorithm exhibited a physical time step dependent

divergent behavior for many problems. This formulation of the algorithm closely followed the development proposed by Jameson [72]. No indication of such behavior was identified in Jameson's [72] paper. Arnone [73] identified a time step modifier which sought to circumvent the unstable region by lowering the pseudo time increment in relation to the physical time step. This modification was originally included in the *ADPAC07* formulation, but did not completely prohibit the instability. Further study of the problem, and, in particular, the assistance of researchers at the NASA-Langley Research Center, have identified the explicit treatment of the physical time derivative term as the source of the conditional stability. In a recent paper by Melson, Sanetrik and Atkins [75], the stability characteristics of the implicit iterative algorithm with both explicit and implicit treatments of the physical time derivative term were analyzed. The implicit treatment of this term was found to be unconditionally stable, while the explicit treatment of the algorithm was conditionally stable based on the value of $\Delta\tau/\Delta t$, where $\Delta\tau$ is the pseudo time derivative time step, while Δt is the physical time derivative time step. Small values of $\Delta\tau/\Delta t$ (corresponding to low CFL numbers) push the algorithm towards an unstable operation. Jameson [72] suggests that CFL numbers of 200 or greater be used for the explicit treatment, which is consistent with low values of $\Delta\tau/\Delta t$, or improved stability. The *ADPAC07* code was therefore modified to utilize an implicit treatment of the time derivative term.

A further improvement of the implicit algorithm was discovered by Melson, et al. This modification is based on the realization that the term Q^{n+1} actually appears on both sides of the implicit time marching equation described above. It should be possible, therefore, to collect these terms and modify the time marching equation in a more "fully implicit" manner (in terms of how the updated Q^{n+1} is calculated). Several variations on this technique are described below.

The development given here follows the derivation described by Melson, et al. Suppose we seek to solve an equation of the form

$$\frac{\partial Q}{\partial t} = L(Q) \quad (3.95)$$

where $L(Q)$ is a collection of fluxes and source terms similar to that given by the Runge-Kutta time marching scheme. The physical time derivative term is approximated by a discrete operator of the form

$$\frac{\partial Q}{\partial t} = \frac{1}{\Delta t} \left(\sum_{m=0}^M a_m Q^{n+1-m} \right) = \frac{1}{\Delta t} [a_0 Q^{n+1} + E(Q^n, Q^{n-1}, \dots, Q^{n+1-m})] \quad (3.96)$$

Here $E(Q)$ represents the portion of the discrete approximation which involves values of the dependent variable Q evaluated from previous time steps.

If we interpret the explicit time marching scheme in the following form:

$$Q^{n+1} = Q^n + \Delta t R(Q) \quad (3.97)$$

where $R(Q)$ is the summation of convective, diffusive, and dissipative fluxes, and internal source terms. For the implicit algorithm, the time step τ becomes a pseudo time step used in an iterative fashion to construct the time dependent solution according to the physical time step ΔT . In this case, the algorithm becomes:

$$Q^{n+1} = Q^n + \Delta\tau \left(\frac{\partial Q}{\partial t} + R(Q) \right) \quad (3.98)$$

If we approximate the physical time derivative term with the discrete operator described above, we get

$$Q^{n+1} = Q^n + \Delta\tau \left(\frac{1}{\Delta t} [a_0 Q^{n+1} + E(Q)] + R(Q) \right) \quad (3.99)$$

and can consequently develop a new implicit-like equation for the term Q^{n+1} as

$$Q^{n+1} = \frac{\left(Q^n + \Delta\tau \left(\frac{1}{\Delta t} [E(Q)] + R(Q) \right) \right)}{1 - \Delta\tau \left(\frac{1}{\Delta t} [a_0] \right)} \quad (3.100)$$

A number of algorithms can now be developed based on the choice of discrete approximation to the physical time derivative term. Based on the work of Melson et al., this study was limited to approximations of both first and second order. Based on linear stability analysis, the algorithm can be made to be unconditionally stable for either first or second order accurate representations of the physical time derivative term. Several higher order approximations were found by Melson et al. to be conditionally stable, and require the added burden of storing more than 3 time levels of data to complete the algorithm.

In every case tested, the modified implicit scheme described by Melson was more robust than the direct implicit scheme described by Arnone. Unfortunately, instabilities were found to occur for both algorithms under conditions which were believed to be in the unconditionally stable regime for each algorithm. It appears that a more thorough investigation of the stability characteristics of the iterative implicit algorithm should be performed to isolate the causes of the intermittent unstable behavior.

3.5 Boundary Conditions

In this section, the various boundary conditions utilized in this study as part of the *ADPAC07* analysis are described. Before describing the individual boundary conditions, it may be useful to describe how the boundary conditions are imposed in the discrete numerical solution. Finite volume solution algorithms such as the *ADPAC07* program typically employ the concept of a phantom cell to impose boundary conditions on the external faces of a particular mesh block. This concept is illustrated graphically for a 2-D mesh representation in Figure 3.10. Some comments concerning the specifics of numerically treating block boundaries are in order at this point. Artificial damping is treated at inflow/outflow block boundaries by prescribing zero dissipation flux along the boundary to preserve the globally conservative nature of the solution. For interblock communication, dissipative fluxes are also communicated between blocks to prevent inadequate numerical damping at inner block boundaries. Implicit residual smoothing is applied at all block boundaries by imposing a zero residual gradient (i.e. $(dR/dz) = 0.0$) condition at the boundary.

A phantom cell is a fictitious neighboring cell located outside the extent of a mesh which is utilized in the application of boundary conditions on the outer boundaries of a mesh block. Since flow variables cannot be directly specified at a mesh surface in a finite volume solution (the flow variables are calculated and stored at cell centers), the boundary data specified in the phantom cells are utilized to control the flux condition at the cell faces of the outer boundary of the mesh block, and, in turn, satisfy a particular boundary condition. All *ADPAC07* boundary condition specifications provide data values for phantom cells to implement a particular mathematical boundary condition *on the mesh*. Another advantage

2-D Mesh Block Phantom Cell Representation

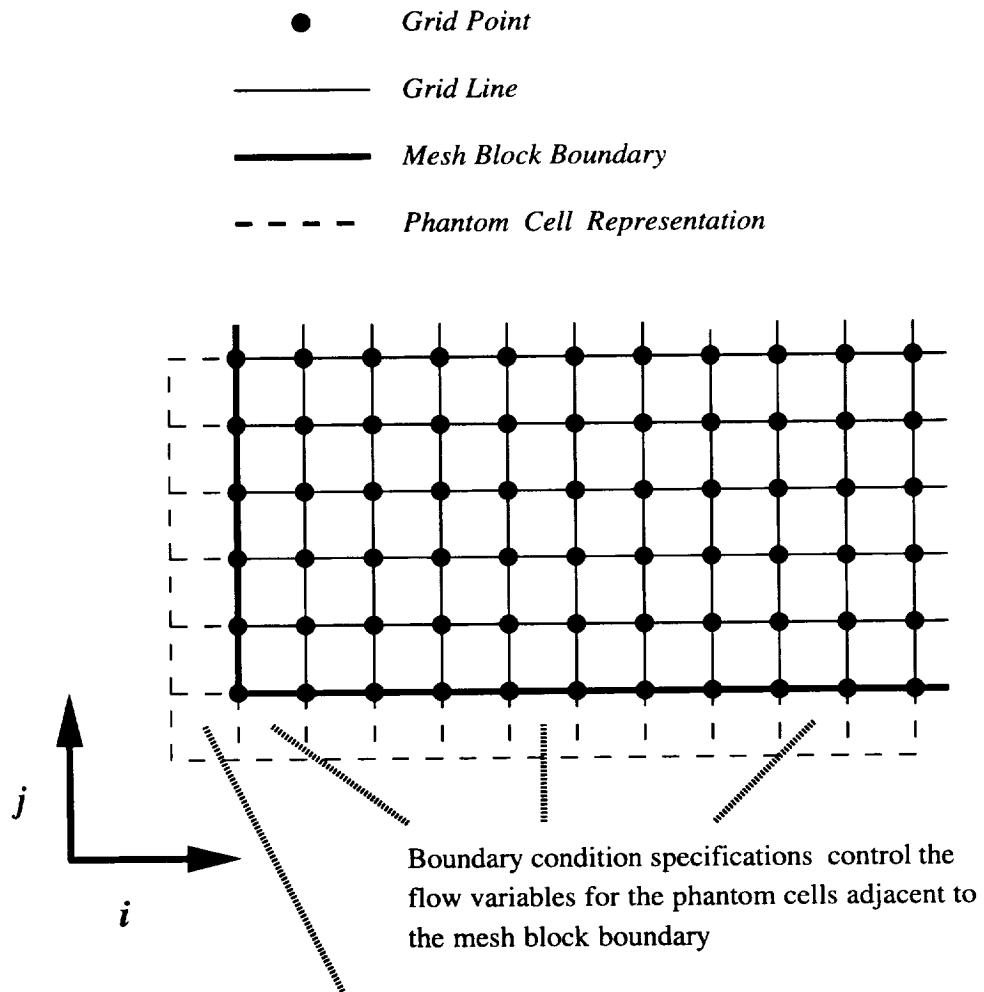


Figure 3.10: 2-D mesh block phantom cell representation.

of the phantom cell approach is that it permits unmodified application of the interior point scheme at near boundary cells.

3.5.1 Standard Inflow/Outflow Boundary Procedures

Inflow and exit boundary conditions are applied numerically using characteristic theory. A one-dimensional isentropic system of equations is utilized to derive the following characteristic equations at an inflow/outflow boundary:

$$\frac{\partial C^-}{\partial t} - (v_n + a) \frac{\partial C^-}{\partial n} = 0, \quad (3.101)$$

$$\frac{\partial C^+}{\partial t} + (v_n + a) \frac{\partial C^+}{\partial n} = 0 \quad (3.102)$$

where:

$$C^- = v_n - \frac{2a}{\gamma - 1}, \quad C^+ = v_n + \frac{2a}{\gamma - 1} \quad (3.103)$$

Numerically, the equations are solved in a locally orthogonal coordinate system which is normal to the cell face of interest (indicated by the subscript and coordinate n). The procedure is essentially then a reference plane method of characteristics based on the Riemann invariants C^- and C^+ .

For subsonic normal inflow, the upstream running invariant C^- is extrapolated to the inlet, and along with the equation of state, specified total pressure, total temperature, and flow angles the flow variables at the boundary may be determined. For turbomachinery flow calculations, the flow angles are representative of the spanwise flow and the pitchwise (blade-to-blade) flow.

Outflow boundaries require a specification of the exit static pressure. In this case, the downstream running invariant C^+ is used to update the phantom cells at the exit boundary. Velocity components parallel to the cell face are extrapolated to the phantom cell from the neighboring interior cells.

It should be mentioned that all of the characteristic boundary schemes utilize a local rotated coordinate system which is normal to the bounding cell face.

3.5.2 Solid Surface Boundary Procedures

All inviscid solid surface must satisfy the condition of no convective flux through the boundary (an impermeable surface). Mathematically, this is expressed as:

$$\vec{V} \cdot \vec{n} = 0 \quad (3.104)$$

The phantom cell velocity components are thus constructed to ensure that the cell face average velocities used in the convective flux calculation satisfy the no throughflow boundary specification *at the bounding surface*. A simplified form of the normal momentum equation is used to update the phantom cell pressure as:

$$\frac{\partial p}{\partial n} = 0 \quad (3.105)$$

It should be noted that this condition is theoretically oversimplified, but our experience using more complicated forms of the normal momentum equation indicate that numerically the results are quite accurate.

All viscous solid surfaces must satisfy the no slip boundary condition for viscous flows:

$$\vec{V}_{rel} = 0 \quad (3.106)$$

where \vec{V}_{rel} is the relative flow velocity, $\vec{V} - r\omega$. No convective flux through the boundary (an impermeable surface) is permitted. The phantom cell velocity components are thus constructed to ensure that the cell face average velocities used in the convective flux calculation are identically zero. The phantom cell pressure is simply extrapolated based on the boundary layer flow concept $dp/dn = 0$. The phantom cell density or temperature is imposed by assuming either an adiabatic surface $dT/dn = 0$ or a specified surface temperature, which suggests that the phantom cell temperature must be properly constructed to satisfy the appropriate average temperature along the surface.

3.5.3 Interblock Communication Boundary Procedures

For the multiple-block scheme, the solution is performed on a single grid block at a time. Special boundary conditions along block boundaries are therefore required to provide some transport of information between blocks. This transport may be accomplished through one of four types of procedures in the *ADPAC07* code. Each procedure applies to a different type of mesh construction and flow environment, and details of each approach are given in Reference [24].

For neighboring mesh blocks which have coincident mesh points along the interface separating the two blocks (as used in this study), a simple direct specification of the phantom cell data based on the near boundary cell data from the neighboring block has been used successfully (**PATCH** boundary condition, see [28]). This concept is illustrated graphically in Figure 3.11. Each phantom cell in the block of interest has a direct correspondance with a near boundary cell in the neighboring mesh block, and the block coupling is achieved numerically by simply assigning the value of the corresponding cell in the neighboring block to the phantom cell of the block of interest. This procedure essentially duplicates the interior point solution scheme for the near boundary cells, and uniformly enforces the conservation principles implied by the governing equations. Other boundary conditions related to interblock communication for endwall treatment flow calculations are described in Chapter 4.

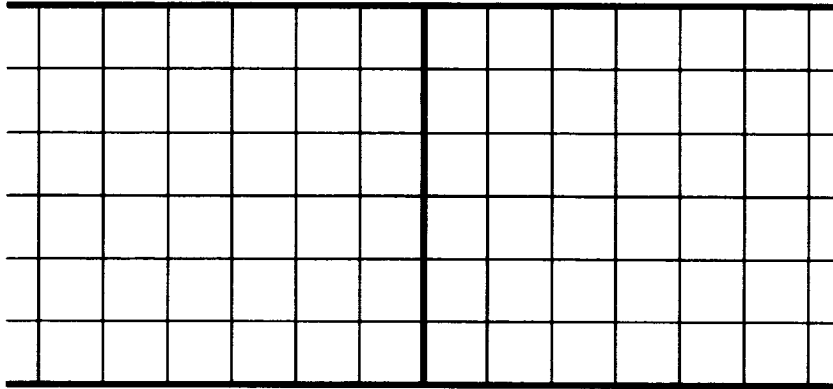
3.5.4 Non-Reflecting Inflow/Outflow Boundary Condition Procedures

A perplexing aspect of the numerical simulation of turbomachinery flowfields is the requirement of specifying inflow/outflow boundary conditions for a limited computational domain for machinery which is operating in an essentially unlimited environment. For the purpose of analyzing these complex flows, our objective is to develop a solution procedure which satisfies three constraints. First, the boundary procedure must maintain a physically consistent far field flow condition which may be specified by the user. Second, the solution should be insensitive to the relative position of the computational inlet and exit boundaries. Third, the boundary conditions should be constructed in a manner which does not introduce spurious, nonphysical reflections of traveling waves into the numerical solution.

Theoretical mathematical foundations for “non-reflecting” boundary conditions for initial value problems can be found in many references ([45], [46] for example). A number of so-called non-reflecting boundary condition procedures have been developed specifically for turbomachinery flow applications have been presented by Erdos et al. [47], among others.

Contiguous Mesh Block Interface Boundary Coupling Scheme

Mesh Block Structure



*Phantom cell data values for mesh block#1
are determined by corresponding near-boundary
data in mesh block #2*

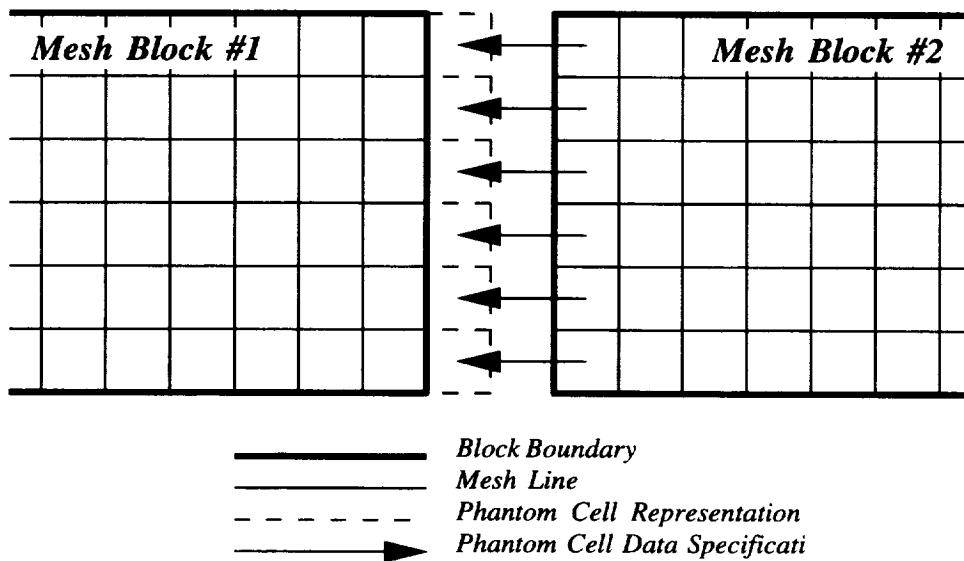


Figure 3.11: ADPAC07 contiguous mesh block coupling scheme.

The boundary condition procedure used in this study follows the general procedure developed by Giles [48], which was later expanded and applied to 3-D time-dependent turbomachinery flow predictions by Saxer [49].

1-D Unsteady Non-Reflecting Boundary Conditions

The development of the nonreflecting boundary condition procedure is best demonstrated by examining the propagation of waves traveling in a single spatial direction. The analysis begins by examining the linearized Euler equations for such a flow written as:

$$\frac{\partial U_p}{\partial t} + A \frac{\partial U_p}{\partial x} = 0 \quad (3.107)$$

In this equation, U_p represents a vector of disturbances to an otherwise uniform flow about which the solution is linearized, and is represented by:

$$U_p = \begin{bmatrix} \rho - \bar{\rho} \\ u - \bar{u} \\ v - \bar{v} \\ w - \bar{w} \\ p - \bar{p} \end{bmatrix} = \begin{bmatrix} \hat{\rho} \\ \hat{u} \\ \hat{v} \\ \hat{w} \\ \hat{p} \end{bmatrix} \quad (3.108)$$

and A is the Jacobian matrix evaluated as:

$$A = \begin{bmatrix} \bar{u} & \bar{\rho} & 0 & 0 & 0 \\ 0 & \bar{u} & 0 & 0 & \frac{1}{\bar{\rho}} \\ 0 & 0 & \bar{u} & 0 & 0 \\ 0 & 0 & 0 & \bar{u} & 0 \\ 0 & \gamma \bar{p} & 0 & 0 & \bar{u} \end{bmatrix} \quad (3.109)$$

Note that the Jacobian matrix \bar{A} is evaluated using variables from the known uniform flow conditions. Saxer [49] notes that these equations represent a local linearization (i.e., within one mesh cell) since the components of \bar{A} can vary in both the radial and circumferential direction for 3-D turbomachinery flow applications.

The matrix \bar{A} can be diagonalized by a similarity transformation of the form

$$T^{-1}AT = \begin{bmatrix} \bar{u} & 0 & 0 & 0 & 0 \\ 0 & \bar{u} & 0 & 0 & 0 \\ 0 & 0 & \bar{u} & 0 & 0 \\ 0 & 0 & 0 & \bar{u} + c & 0 \\ 0 & 0 & 0 & 0 & \bar{u} - c \end{bmatrix} = \Lambda \quad (3.110)$$

Here $c = \sqrt{\frac{\gamma p}{\rho}}$ is the mean flow speed of sound. The diagonal components of Λ represent the speed of propagation of of five characteristic waves. Multiplication of Equation(3.107) by T yields

$$\frac{\partial \Phi}{\partial t} + \Lambda \frac{\partial \Phi}{\partial x} = 0 \quad (3.111)$$

where $\Phi = T^{-1}U_p$ is referred to as the vector of linearized characteristic variables, and, in detail, are given by

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} -\bar{c}^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \bar{\rho}\bar{c} & 0 & 0 \\ 0 & 0 & 0 & \bar{\rho}\bar{c} & 0 \\ 0 & \bar{\rho}\bar{c} & 0 & 0 & 1 \\ 0 & -\bar{\rho}\bar{c} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\rho} \\ \hat{u} \\ \hat{v} \\ \hat{w} \\ \hat{p} \end{bmatrix} \quad (3.112)$$

The corresponding transformation from characteristic to primitive variables is given by

$$\begin{bmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{p} \end{bmatrix} = \begin{bmatrix} \frac{-1}{c^2} & 0 & 0 & \frac{1}{2\tilde{c}^2} & \frac{1}{2\tilde{c}^2} \\ 0 & 0 & 0 & \frac{1}{2\tilde{c}^2} & \frac{-1}{2\tilde{c}^2} \\ 0 & \frac{1}{\rho c} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\rho c} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} \quad (3.113)$$

By interrogating the direction associated with the linearized characteristic variables, the correct non-reflecting boundary conditions for a subsonic normal inflow boundary may be expressed as:

$$\begin{aligned} \phi_1 &= 0 \\ \phi_2 &= 0 \\ \phi_3 &= 0 \\ \phi_4 &= 0 \end{aligned} \quad (3.114)$$

and for a subsonic normal exit flow boundary as:

$$\phi_5 = 0 \quad (3.115)$$

Numerically, the boundary condition algorithm is implemented by calculating or extrapolating the outgoing characteristic variables from the interior domain, and using Equation(3.113) to reconstruct the solution on the boundary.

The non-reflecting boundary conditions for *ADPAC07* were reformulated from the original equations for the 1-D unsteady non-reflecting boundary conditions. A simple 2-D channel with a uniform pressure disturbance wave (1-D problem) was studied using both the generic (reflective) boundary conditions and the newly-formulated non-reflecting boundary conditions. Time-accurate solutions were obtained for two grid sizes both with the same grid resolution; the longer grid spanned from -1 to 2 in the x-direction and the shorter grid, a subset of the longer grid, only spanned from 0 to 1. Pressure histories were gathered for a pressure wave traveling both in a quiescent environment and in a Mach 0.67 flow. These pressure traces were compared at various time intervals. The resulting pressure trace histories, shown in Figures 3.12 and 3.13, show a remarkable reduction in reflected waves using the non-reflecting boundaries. In both flow cases, the generic (reflective) boundary conditions generate false waves which continue to bounce between the inlet and exit boundaries long after the solution should have come to rest; whereas, the non-reflecting boundary conditions appear to let the pressure waves pass through the boundaries with minimal reflection.

Solutions were obtained for an oscillating flat plate cascade using a series of grids with decreasing distance between the cascade and the boundaries. This distance ranged from six chords away down to one-quarter chord away. The *ADPAC07* solutions from these grids are shown in comparison to the Smith linearized data in Figure 3.14. Also included are the results using the *ADPAC07* 1-D unsteady non-reflecting boundary conditions on the shortest grid. These solutions were run through 25 cycles with a semi-chord reduced frequency of 4.0, a freestream Mach number of 0.5, and an interblade phase angle of 0. As the boundaries were brought closer to the cascade, the accuracy of the predicted solutions did not degrade as was expected from previous studies.

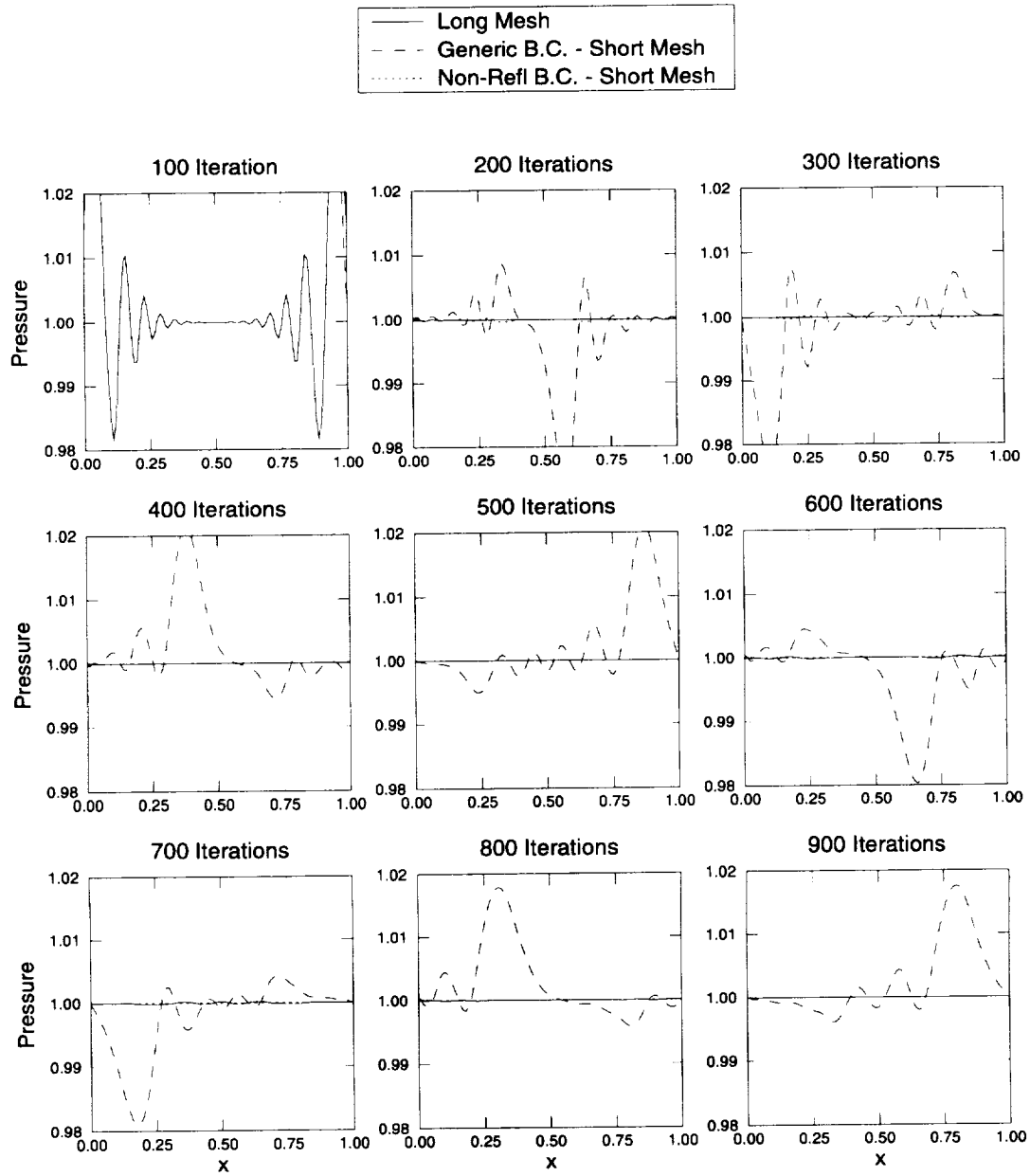


Figure 3.12: Time history of pressure traces for a 1-D pressure wave traveling in a Mach 0.00 (stationary) flow, demonstrating the advantages of non-reflecting boundary conditions.

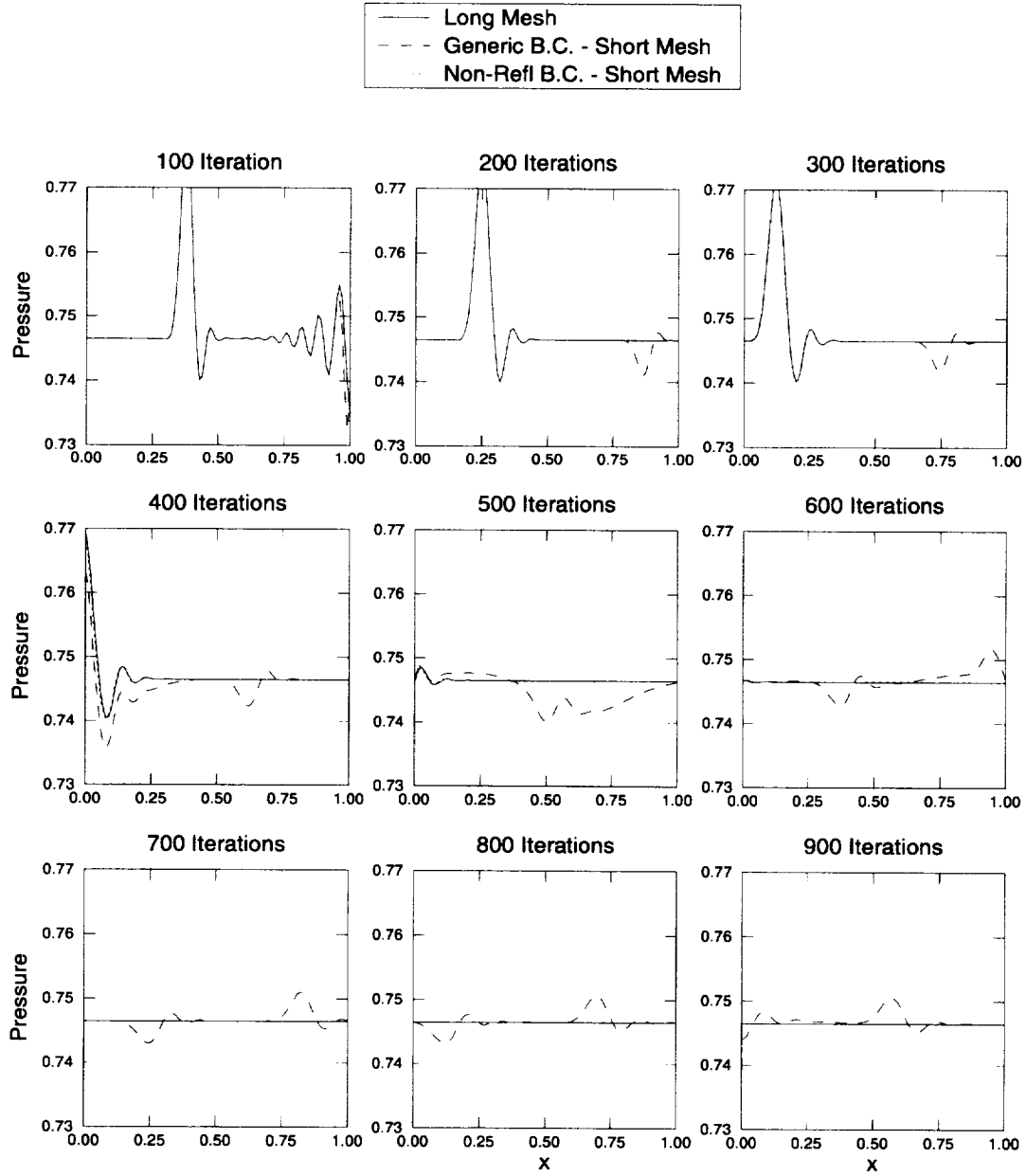


Figure 3.13: Time history of pressure traces for a 1-D pressure wave traveling in a Mach 0.67 flow, demonstrating the advantages of non-reflecting boundary conditions.

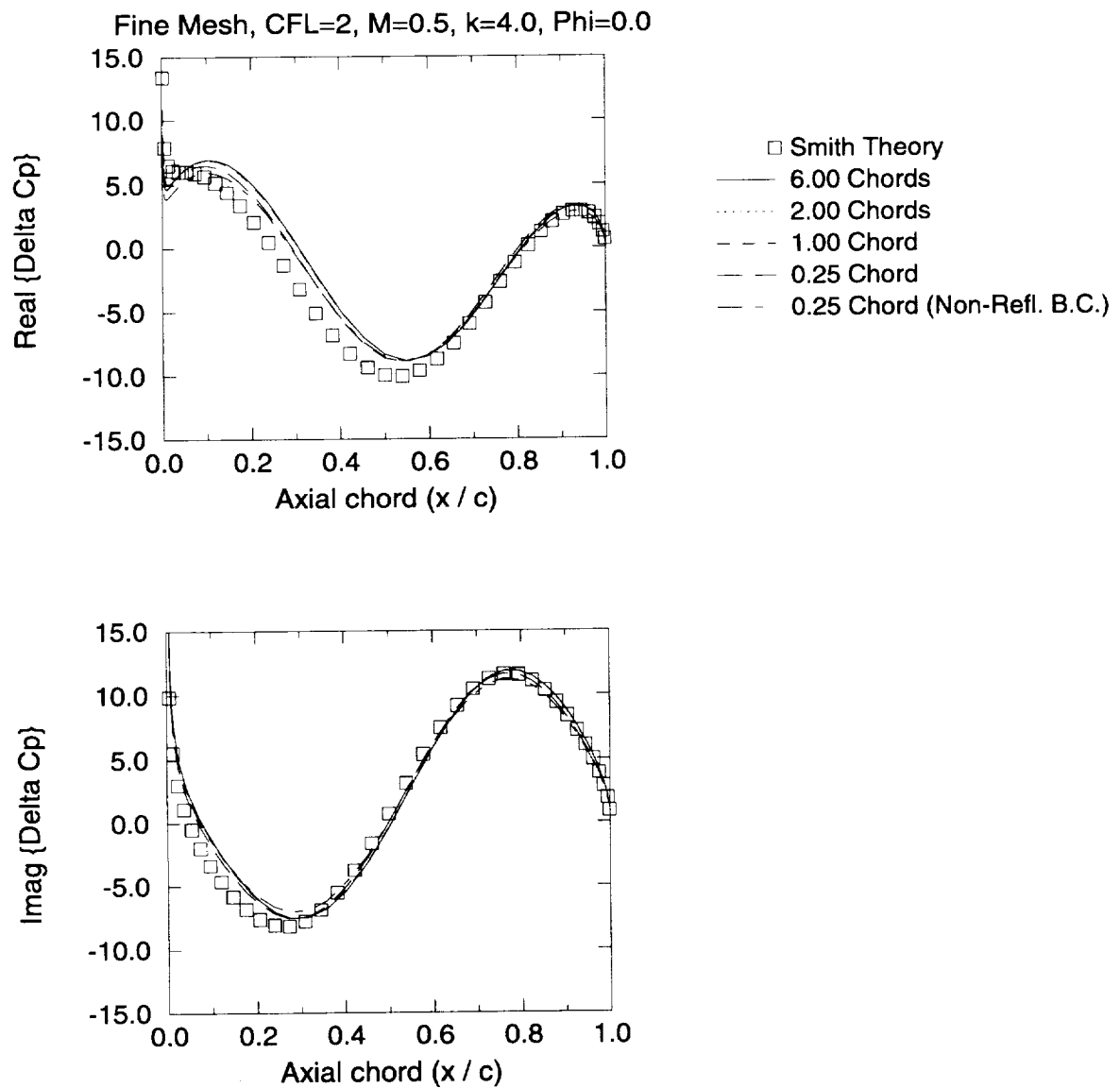


Figure 3.14: Comparison of *ADPAC07* results for an oscillating flat plate cascade with the Smith linear theory examining the effect of shortening the distance to the computational boundaries and the effect of a non-reflecting boundary condition on the shortest mesh.

Extension to 3-D Flow

The extension from the 1-D non-reflecting boundary conditions, described above, to higher dimensions has taken the form of adapting Saxer's quasi-3-D steady boundary conditions to *ADPAC07*. These quasi-3-D boundary conditions are applied by using the 2-D non-reflecting boundaries at each constant radial slice. Saxer's boundary conditions use a Fourier decomposition in the pitchwise direction to determine the correct incoming characteristics in terms of the outgoing characteristics. The characteristics are related to perturbations in the primitive variables through the equation shown below:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} -c^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \bar{\rho}\bar{c} & 0 & 0 \\ 0 & 0 & 0 & \bar{\rho}\bar{c} & 0 \\ 0 & \bar{\rho}\bar{c} & 0 & 0 & 1 \\ 0 & -\bar{\rho}\bar{c} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (\rho - \bar{\rho}) \\ (u_x - \bar{u}_x) \\ (u_\theta - \bar{u}_\theta) \\ (u_r - \bar{u}_r) \\ (p - \bar{p}) \end{bmatrix}$$

The details of the implementation into the *ADPAC07* code are outlined below following the derivations by Saxer [49] and Giles [48]. 2-D non-reflecting boundary conditions are applied at each radial slice in the 3-D grid designated below by the index j ranging on the mesh from j_1 to j_2 and the circumferential index by k ranging from k_1 to k_2 .

The first step at each radial slice begins by finding the average flow quantities about which to linearize. This flux-averaging process is shown below with the blade pitch defined by P :

$$\bar{F} = \frac{1}{P} \sum_{k=k_1+1}^{k_2} F_k \Delta\theta_k$$

where,

$$F_1 = \rho u_x$$

$$F_2 = \rho u_x^2 + p$$

$$F_3 = \rho u_x u_\theta$$

$$F_4 = \rho u_x u_r$$

$$F_5 = \rho u_x I$$

$$I = \frac{\gamma}{\gamma - 1} \frac{p}{\rho} + \frac{1}{2} (u_x^2 + u_\theta^2 + u_r^2) - \frac{1}{2} \Omega^2 r^2$$

The averaged primitive variables can be found from the above F array through the relationships listed below:

$$p = \frac{1}{\gamma + 1} \left(F_2 + \sqrt{\bar{F}_2^2 + (\gamma^2 - 1)(F_2^2 + \bar{F}_3^2 + \bar{F}_4^2 - 2F_1 F_5 - (F_1 \Omega r)^2)} \right)$$

$$u_x = \frac{F_2 - \bar{p}}{F_1}$$

$$u_\theta = \frac{F_3}{F_1}$$

$$u_r = \frac{\bar{F}_4}{F_1}$$

$$\rho = \frac{\bar{F}_1}{\bar{u}_x}$$

Subsonic Inlet

These averaged flow variables are then used to construct the four residual equations used to enforce the subsonic inlet conditions. The equations have been implemented into *AD-PAC07* using the original specified variables $(s, \alpha_\theta, \alpha_r, h_t)$ used by Saxer and are shown below followed by the transformation used to convert the standard *ADPAC07* inlet specified variables $(p_t, T_t, \beta_r, \beta_\theta)$ to those used in the residual equations.

$$R_1 = p(s - s_i)$$

$$R_2 = \rho c \left(u_\theta - \left| \vec{V} \right| \sin(\alpha_\theta) \sin(\alpha_r) \right)$$

$$R_3 = \rho c \left(\bar{u}_r - \left| \vec{V} \right| \cos(\alpha_r) \right)$$

$$R_4 = \bar{\rho}(h_t - h_{t_i})$$

where,

$$s_i = \ln(\gamma p_t) - \gamma \ln(\rho_t)$$

$$\alpha_\theta = -\beta_\theta$$

$$\alpha_r = \arctan \left(\frac{1}{\cos(\beta_\theta) \tan(\beta_r)} \right)$$

$$h_{t_i} = c_p T_t$$

Using a one-step Newton-Raphson procedure, the residuals can be related to the average change in the four characteristics. The inverse relation can be found using the inverted matrix of the Jacobian shown below:

$$[J] = \frac{\partial(R_1, R_2, R_3, R_4)}{\partial(\phi_1, \phi_2, \phi_3, \phi_4)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{\tan(\alpha_\theta)}{2} \\ 0 & 0 & 1 & -\frac{1}{2 \cos(\alpha_\theta) \tan(\alpha_r)} \\ \frac{1}{\gamma-1} & M_\theta & M_r & \frac{1}{2}(1 + M_r) \end{bmatrix}$$

where,

$$M_{x'} = \frac{\bar{u}_x}{c} \quad M_r = \frac{u_r}{c} \quad M_\theta = \frac{\bar{u}_\theta}{c}$$

The average global change in the incoming characteristics is related to the residuals as:

$$\begin{bmatrix} \delta\hat{\phi}_1 \\ \delta\hat{\phi}_2 \\ \delta\hat{\phi}_3 \\ \delta\phi_4 \end{bmatrix} = -[J^{-1}] \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix}$$

where,

$$[J^{-1}] = \frac{1}{1 + M_x + M_\theta \tan(\alpha_\theta) + \frac{M_r}{C'T}} [\tilde{J}]$$

and

$$[\tilde{J}] = \begin{bmatrix} 1 + M_x + M_\theta \tan(\alpha_\theta) + \frac{M_r}{C'T} & 0 & 0 & 0 \\ -\frac{1}{\gamma-1} \tan(\alpha_\theta) & 1 + M_x + \frac{M_r}{C'T} & -M_r \tan(\alpha_\theta) & \tan(\alpha_\theta) \\ -\frac{1}{(\gamma-1)C'T} & \frac{M_\theta}{C'T} & 1 + M_x + M_\theta \tan(\alpha_\theta) & \frac{1}{C'T} \\ -\frac{2}{\gamma-1} & -2M_\theta & -2M_r & 2 \end{bmatrix}$$

and

$$C'T = \cos(\alpha_\theta) \tan(\alpha_r)$$

There are a few special cases, primarily those where the flow angle definitions go to 0° or 90° , where the above matrix simplifies or becomes undefinable. Tests for these special cases have been coded into *ADPAC07* and are treated accordingly.

Once the global changes in the incoming characteristics for the radial slice are found, the local change must be determined through a Fourier transform. The complex variables used in this derivation are symbolized by hats ($\hat{\phi}$). As mentioned above, the incoming local characteristics are derived from the outgoing characteristics. For the subsonic inlet case, the only outgoing characteristic is ϕ_5 and its Fourier transform is found by:

$$\hat{\phi}_{5n} = \frac{1}{P} \sum_{k=k_1+1}^{k_2} \phi_{5k} \exp\left(\frac{-i2\pi n(\theta_k - \theta_{k_1})}{P}\right) \Delta\theta_k$$

for $n = 1$ to $(k_2 - k_1)/2 - 1$. One of the incoming characteristics can be defined in terms of $\hat{\phi}_5$ by:

$$\hat{\phi}_{2n} = -\frac{\beta + u_\theta}{c + \hat{u}_x} \hat{\phi}_{5n}$$

where for the subsonic case,

$$\beta = i \frac{n}{\|n\|} \sqrt{c^2 - (\hat{u}_x^2 + u_\theta^2)}$$

By using an inverse Fourier transform, the local steady-state second incoming characteristic is found at each circumferential location by:

$$\phi_{2ks} = 2\Re \left\{ \sum_{n=1}^{(k_2-k_1)/2-1} \hat{\phi}_{2n} \exp \left(\frac{i2\pi n(\theta_k - \theta_{k_1})}{P} \right) \right\}$$

The local change in the characteristic can then be found by differencing the steady-state value and the current value:

$$\delta\phi_{2ks} = \phi_{2ks} - \phi_{2k}$$

The third incoming characteristic harmonics are zero, therefore the local change in the third characteristic is:

$$\delta\phi_{3ks} = -\phi_{3k}$$

Corrections need to be made to the first and fourth characteristics such that the local values of entropy and total enthalpy match the average values. These values can be obtained through a similar Newton-Raphson procedure used earlier in the derivation of the global changes. The residuals for the local entropy and total enthalpy are set up as shown below:

$$R_{1k} = p(s_k - \bar{s})$$

$$R_{4k} = \bar{\rho}(h_{tk} - h_t)$$

From the Newton-Raphson procedure, the local change in the remaining characteristics can be found as:

$$\delta\phi_{1ks} = -R_{1k}$$

$$\delta\phi_{4ks} = -\frac{2}{1 + M_x} \left(\frac{1}{\gamma - 1} \delta\phi_{1ks} + M_\theta \delta\phi_{2ks} + M_r \delta\phi_{3ks} + R_{4k} \right)$$

Once all the global and local changes have been determined the cumulative change can be found, multiplied by an under-relaxation factor currently set by $\sigma = 1/(k_2 - k_1)$:

$$\delta\phi_{1k} = \sigma(\delta\phi_1 + \delta\phi_{1ks})$$

$$\delta\phi_{2k} = \sigma(\delta\phi_2 + \delta\phi_{2ks})$$

$$\delta\phi_{3k} = \sigma(\delta\phi_3 + \delta\phi_{3ks})$$

$$\delta\phi_{4k} = \sigma(\delta\phi_4 + \delta\phi_{4ks})$$

The change to the fifth (outgoing) characteristic is calculated from the interior domain by:

$$\delta\phi_{5k} = -\bar{\rho}\bar{c}(u_x - u_x) + (p - p)$$

Supersonic Inlet

For the supersonic inflow, $(\bar{u}_x^2 + \bar{u}_\theta^2) > \bar{c}^2$, the calculation for the global change remains unchanged and the Fourier transform is no longer required. The steady-state value for the second characteristic can now be defined as:

$$\phi_{2ks} = -\frac{\beta + \bar{u}_\theta}{\bar{c} + \bar{u}_x} \phi_{5k}$$

where β is now defined as:

$$\beta = -\text{sign}(\bar{u}_\theta) \sqrt{(\bar{u}_x^2 + \bar{u}_\theta^2) - \bar{c}^2}$$

Once the steady-state value is found, the remainder of the boundary conditions remain as described above.

Subsonic Exit

At the exit, the first four characteristics are determined from the interior of the domain, and only the fifth characteristic is incoming. The equation used to determine the global change in the fifth characteristic is:

$$\delta\phi_5 = -2(p - \bar{p}_{exit})$$

where, \bar{p}_{exit} is the specified average static pressure at the radial slice which the boundary condition is being applied. The distribution of the static pressure is determined through satisfying radial equilibrium at the exit.

To determine the local change in the incoming characteristic, discrete Fourier transforms of two of the outgoing characteristics are required:

$$\hat{\phi}_{2n} = \frac{1}{P} \sum_{k=k_1+1}^{k_2} \phi_{2k} \exp\left(\frac{-i2\pi n(\theta_k - \theta_{k_1})}{P}\right) \Delta\theta_k$$

$$\hat{\phi}_{4n} = \frac{1}{P} \sum_{k=k_1+1}^{k_2} \phi_{4k} \exp\left(\frac{-i2\pi n(\theta_k - \theta_{k_1})}{P}\right) \Delta\theta_k$$

for $n = 1$ to $(k_2 - k_1)/2 - 1$. The incoming characteristics can be defined in terms of $\hat{\phi}_2$ and $\hat{\phi}_4$ by:

$$\hat{\phi}_{5n} = \frac{2\bar{u}_x}{\beta - \bar{u}_\theta} \hat{\phi}_{2n} - \frac{\beta + \bar{u}_\theta}{\beta - \bar{u}_\theta} \hat{\phi}_{4n}$$

where β is defined in the subsonic inlet section above. The steady-state value of the fifth characteristic can be found through the following relation:

$$\phi_{5ks} = 2\Re \left\{ \sum_{n=1}^{(k_2-k_1)/2-1} \hat{\phi}_{5n} \exp\left(\frac{i2\pi n(\theta_k - \theta_{k_1})}{P}\right) \right\}$$

The local change in the characteristic is then found by differencing the steady-state value and the local value:

$$\delta\phi_{5ks} = \phi_{5ks} - \phi_{5k}$$

The total change in the fifth characteristic is calculated by summing the local and global changes then multiplying by an under-relaxation factor:

$$\delta\phi_{5k} = \sigma(\delta\phi_5 + \delta\phi_{5ks})$$

The change in the remaining four (outgoing) characteristics is determined from the interior:

$$\delta\phi_{1k} = -c^2(\rho - \bar{\rho}) + (p - \bar{p})$$

$$\delta\phi_{2k} = \bar{\rho}\bar{c}(u_\theta - \bar{u}_\theta)$$

$$\delta\phi_{3k} = \rho\bar{c}(u_r - \bar{u}_r)$$

$$\delta\phi_{4k} = \bar{\rho}\bar{c}(u_r - \bar{u}_r) + (p - \bar{p})$$

Supersonic Exit

The only change to the exit procedure above if the exit flow is supersonic is the Fourier transform is no longer required and the steady-state value of the fifth characteristic is found by:

$$\phi_{5ks} = \frac{2\bar{u}_r}{\beta - \bar{u}_\theta}\phi_{2k} - \frac{\beta + u_\theta}{\beta - \bar{u}_\theta}\phi_{4k}$$

where β is defined in the supersonic inlet section above. The remainder of the exit boundary conditions follows the same derivation as for subsonic exits described above.

Once the total changes to the characteristic variables have been determined from the appropriate boundary condition above, the corresponding change in the primitive variables can be found by:

$$\begin{bmatrix} \delta\rho \\ \delta u_r \\ \delta u_\theta \\ \delta u_r \\ \delta p \end{bmatrix} = \begin{bmatrix} -\frac{1}{c^2} & 0 & 0 & \frac{1}{2c^2} & \frac{1}{2c^2} \\ 0 & 0 & 0 & \frac{1}{2\rho c} & \frac{1}{2\rho c} \\ 0 & \frac{1}{\rho c} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\rho c} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \delta\phi_1 \\ \delta\phi_2 \\ \delta\phi_3 \\ \delta\phi_4 \\ \delta\phi_5 \end{bmatrix}$$

With the primitive variables updated, the conservative variables are set in the phantom cells to enforce these new variables at the boundary.

One of the features of using these boundary conditions is that rather than enforcing a constant boundary value at every point along the pitch, the values may vary circumferentially as long as the average flow value is equal to the prescribed boundary condition. This means in practice the upstream and downstream boundaries can be brought in closer, thereby reducing the number of grid points needed, without degrading the solution. Some initial test cases indicated that the Saxer boundary conditions also accelerate convergence of the solution; this is probably related to the reduction in spurious waves bouncing back and forth between the inlet and exit boundaries.

Residual Equations Reformulation

The majority of the details for the implementation of the non-reflecting boundary conditions into the *ADPAC07* code was outlined in the paragraphs above following the derivations by Saxer [49] and Giles [48]. However; some changes were made to the derivation during the code debugging phase to facilitate a more direct application for turbomachinery flow calculations. These minor changes are discussed below.

After the flow variables are averaged in the pitchwise direction, these quantities are used to construct the four residual equations used to enforce the subsonic inlet conditions. The residual equations have been reformulated using the *ADPAC07* user specified variables for total pressure, total temperature, radial flow angle, and tangential flow angle ($p_{t,m}$, $T_{t,m}$, $\beta_{r,m}$, $\beta_{\theta,m}$) rather than those presented by Saxer. The new residual equations are listed below:

$$\begin{aligned} R_1 &= p_t - p_{t,m} = \bar{p} \left[1 + \frac{\rho \bar{V}^2}{2c_p p} \right]^{\frac{\gamma}{\gamma-1}} - p_{t,m} \\ R_2 &= u_\theta - u_x \tan(\beta_{\theta,m}) \\ R_3 &= u_r - u_x \tan(\beta_{r,m}) \\ R_4 &= T_t - T_{t,m} = \left[\frac{\bar{p}}{\rho} + \frac{\bar{V}^2}{2c_p} \right] - T_{t,m} \end{aligned}$$

The residuals are related to the average change in the four characteristics using a one-step Newton-Raphson procedure. The changes in the residual equations were taken into account when recalculating the inverse Jacobian used to determine the global changes in the four characteristics at the inlet.

A change was also made in the under-relaxation factor used with the total sum of the characteristics. The under-relaxation now only modifies the value of the local characteristic change and not the sum of the local and global values. This change appears to be a more direct application of the theory presented in Saxer's thesis.

Sample Turbomachinery Application

One of the advantages of using these boundary conditions is that rather than enforcing a constant boundary value at every point along the pitch, the values may vary circumferentially as long as the average flow value is equal to the prescribed boundary condition. This means in practice the upstream and downstream boundaries can be brought in closer, thereby reducing the number of grid points needed, without degrading the solution.

The NASA Rotor 67 was selected as a test case to determine the effects of the non-reflecting boundary conditions. Inviscid calculations were performed for the rotor using an H-type mesh with boundaries located away from the blade using the original *ADPAC07* turbomachinery boundary conditions. To show the influence of the mesh boundary, the inlet and exit boundary locations were later brought in extremely close to the blade. The locations of these boundaries on the shortened mesh were taken to an extreme to amplify the effect of the boundaries on the solution. Inviscid solutions were gathered using the shortened mesh for both the original turbomachinery boundary conditions and the new non-reflecting boundary conditions.

The static pressure distributions at the casing surface were compared for the three solutions. Figure 3.15 show the results from the original long mesh along with an outline

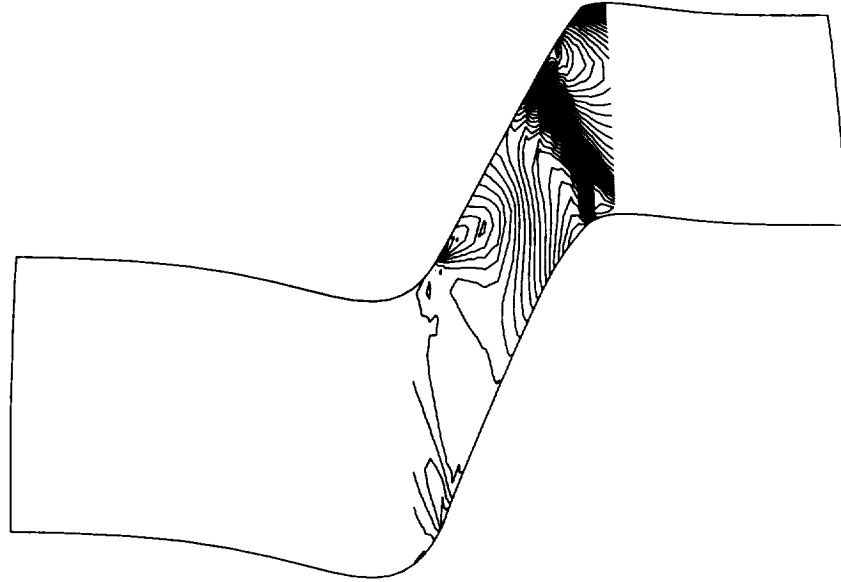


Figure 3.15: Static pressure contours at the casing for Rotor 67 on the long grid using the original turbomachinery boundary conditions.

of the mesh boundary. The contour lines are clipped at the same location from which the short mesh was extracted from the original mesh for ease of comparison. Figures 3.16 and 3.17 show the results from the shortened mesh using the original turbomachinery boundary conditions and the non-reflecting boundary conditions, respectively. The specification of constant static pressure along the pitch at the exit can be seen in Figure 3.16 as the contour lines are compressed back into the solution domain. In comparison, the non-reflecting boundary conditions results, shown in Figure 3.17, allow for variations across the pitch which can be seen as the static pressure contour lines pass through the boundary. From these (and other) results, the non-reflecting boundary condition calculation on the short mesh appears to match more closely the original long mesh solution.

In terms of computational expense, the use of the non-reflecting boundary procedures certainly requires additional CPU time compared to the simple Riemann invariant boundary procedures. The added CPU cost, while not measured directly, was believed to be quite small. One could argue that this added computational effort could conceivably have been used to extend the mesh (placing the boundaries farther away) and employ simpler (and cheaper) boundary procedures with similar results. The drawback to this approach is that the trade-off point between additional mesh points and boundary condition expense is difficult to define on a case-by-case basis, and evaluation of the effect of boundary placement on solution accuracy almost certainly requires multiple calculations for a given solution. Use of the non-reflecting boundary procedures eliminates these questions and serves to remove at least one potential source of solution inaccuracy.

3.6 Turbulence Model

As a result of computer limitations regarding storage and execution speed, the effects of turbulence are introduced through an appropriate turbulence model and solutions are per-

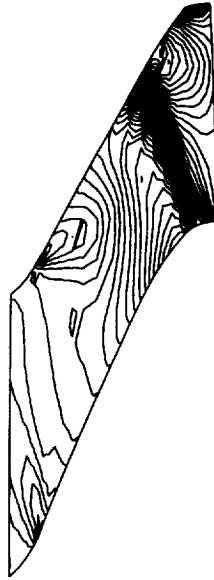


Figure 3.16: Static pressure contours at the casing for Rotor 67 on the short grid using the original turbomachinery boundary conditions.

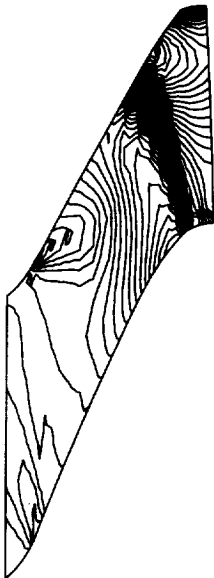


Figure 3.17: Static pressure contours at the casing for Rotor 67 on the short grid using the new non-reflecting boundary conditions.

formed on a numerical grid designed to capture the macroscopic (rather than the microscopic) behavior of the flow.

The effects of turbulence are introduced into the numerical scheme by utilizing the Boussinesq approximation (see e.g. [42]), resulting in an effective calculation viscosity defined as:

$$\mu_{effective} = \mu_{laminar} + \mu_{turbulent} \quad (3.116)$$

The simulation is therefore performed using an effective viscosity which combines the effects of the physical (laminar) viscosity and the effects of turbulence through the turbulence model and the turbulent viscosity $\mu_{turbulent}$. The turbulent flow thermal conductivity term is also treated as the combination of a laminar and turbulent quantity as:

$$k_{effective} = k_{laminar} + k_{turbulent} \quad (3.117)$$

For turbulent flows, the turbulent thermal conductivity $k_{turbulent}$ is determined from a turbulent Prandtl number $Pr_{turbulent}$ such that

$$Pr_{turbulent} = \frac{c_p \mu_{turbulent}}{k_{turbulent}} \quad (3.118)$$

The turbulent Prandtl number is normally chosen to have a value of 0.9.

In this study, two different types of turbulence model were used to compute the eddy viscosity used in the Boussinesq approximation described above. The first model is referred to as an algebraic turbulence model due to the algebraic nature by which the turbulent viscosity is calculated. Algebraic models are generally the simplest models available for computational aerodynamic analysis, and are “tuned” based on correlations with flat plate turbulent boundary layers. Unfortunately, the simplicity of the modeling approach limits the useful applicability of the model to flows which consist primarily of well behaved (non-separated) wall bounded shear layers. To overcome this limitation, a two-equation turbulence model was constructed based on the turbulence kinetic energy and turbulence Reynolds number. Two equation models generally overcome the limitations of algebraic models, but require substantially greater coding and computer resources to implement. Both models are described in greater detail in the sections which follow.

3.6.1 Algebraic (Baldwin-Lomax) Turbulence Model

A relatively standard version of the Baldwin-Lomax [34] turbulence model was adopted for the algebraic model used in the *ADPAC07* analysis. This model is computationally efficient, and has been successfully applied to a wide range of geometries and flow conditions. The Baldwin-Lomax model specifies that the turbulent viscosity be based on an inner and outer layer of the boundary layer flow region as:

$$\mu_{turbulent} = \begin{cases} (\mu_{turbulent})_{inner}, & y \leq y_{crossover} \\ (\mu_{turbulent})_{outer}, & y > y_{crossover} \end{cases} \quad (3.119)$$

where y is the normal distance to the nearest wall, and $y_{crossover}$ is the smallest value of y at which values from the inner and outer models are equal. The inner and outer model turbulent viscosities are defined as:

$$(\mu_{turb})_{inner} = \rho l^2 |\omega| \quad (3.120)$$

$$(\mu_{turb})_{outer} = K C'_{cp} \rho F_{wake} F_{kleb} y \quad (3.121)$$

Here, the term l is the Van Driest damping factor

$$l = \kappa y(1 - e^{(-y^+/A^+)}) \quad (3.122)$$

ω is the vorticity magnitude, F_{wake} is defined as:

$$F_{wake} = y_{max} F_{max} \quad (3.123)$$

where the quantities y_{max} , F_{max} are determined from the function

$$F(y) = y|\omega|[1 - e^{(-y^+/A^+)})] \quad (3.124)$$

The term y^+ is defined as

$$y \left(\sqrt{\frac{\rho|\omega|}{\mu_{laminar}}} \right)_{wall} \quad (3.125)$$

The quantity F_{MAX} is the maximum value of $F(y)$ that occurs across the boundary layer profile, and y_{MAX} is the value of y at which $F_{MAX}(y)$ occurs. The determination of F_{MAX} and y_{MAX} is perhaps the most difficult aspect of this model for three-dimensional flows. The profile of $F(y)$ versus y can have several local maximums, and it is often difficult to establish which values should be used. In this case, F_{MAX} is taken as the maximum value of $F(y)$ between a y^+ value of 100.0 and 1200.0. The function F_{kleb} is the Klebanoff intermittency factor given by

$$F_{kleb}(y) = [1 + 5.5(\frac{C'_{kleb}y}{y_{max}})^6]^{-1} \quad (3.126)$$

and the remainder of the terms are constants defined as:

$$A^+ = 26,$$

$$C'_{cp} = 1.6,$$

$$C'_{kleb} = 0.3,$$

$$\kappa = 0.4,$$

$$K = 0.0168 \quad (3.127)$$

In practice, the turbulent viscosity is limited such that it never exceeds 1000.0 times the laminar viscosity.

In order to properly utilize this turbulence model, a fairly large number of grid cells must be present in the boundary layer flow region, and, perhaps of greater importance, the spacing of the first grid cell off of a wall should be small enough to accurately account for the inner “law of the wall” turbulent boundary layer profile region ($y^+ \leq 5$). Unfortunately, this constraint is often not satisfied due to grid-induced problems or excessive computational costs. Special attention was given to the problems associated with grid refinement and the resulting effects on predicted heat transfer in Reference [25].

Practical applications of the Baldwin-Lomax model for three-dimensional viscous flow must be made with the limitations of the model in mind. The Baldwin-Lomax model was designed for the prediction of wall bounded turbulent shear layers, and is not likely to be well suited for flows with massive separations or large vortical structures. There are, unfortunately, a number of applications for turbomachinery where this model is likely to be invalid.

3.6.2 Wall Functions

The Baldwin-Lomax turbulence model currently implemented in the *ADPAC* code is valid for many flows of engineering interest provided that adequate mesh resolution is available to capture the subscale, near-wall viscous flow behavior which is crucial to correctly predict the overall boundary layer flow characteristics. For most applications, this implies that the first mesh point away from the wall must be located at a value of y^+ less than or equal to 1.0, where y^+ is defined as:

$$y^+ = \frac{y\rho_{wall}}{\mu_{wall}} \sqrt{\frac{\tau_{wall}}{\rho_{wall}}}$$

where τ_{wall} represents the viscous shear stress at the wall, and μ_{wall} and ρ_{wall} are the fluid viscosity and density at the wall. Unfortunately, for many cases it is not possible or feasible to comply with this restriction due to tradeoffs between minimizing both the overall number of grid points and mesh stretching ratios. Additional computational considerations must be given for time-dependent flows, where the maximum allowable time step is often dictated by the near-wall mesh spacing. The wall function method is widely used as an approach towards resolving the influence of the near-wall flow behavior without actually discretizing the inner portion of the boundary layer flow. The wall function method is, quite simply, an empirical specification of the wall shear stress based on local near-wall flow characteristics. This approach has several advantages including a computational savings (both CPU time and memory), and by providing a means by which additional empirical information about a particular flow may be introduced to the numerical solution (e.g. surface roughness modeled by a modified wall shear stress relation) at little or no additional cost. Wall function techniques for turbulence models have been proposed and used by many authors including Spalding [64], Wolfshtein [65], Patankar and Spalding [66], and Launder and Spalding [67].

The implementation of the wall function procedure in the *ADPAC* code is based on a rather novel approach involving the manipulation of the near-wall eddy viscosity. The “standard” method for implementing wall functions is to relax the no-slip wall boundary condition (allow a slip velocity at the wall) based on the requirement of no normal flow and the specified wall shear stress. This approach often requires specific modifications to the turbulence model, near-wall boundary conditions, viscous stress calculation, and energy equation solution routines. The finite volume formulation utilized in the *ADPAC* code allows for a number of options when implementing the wall functions formula. An illustration of the near-wall computational configuration for the *ADPAC* code is given in Figure 3.18.

The objective during the *ADPAC* wall function implementation was to minimize the number of routines which required modification in order to implement the wall function model. This goal suggested that it was desirable to maintain the no-slip wall boundary condition in its original form, and hence the specified wall shear stress was implemented by manipulating the value of the phantom cell turbulent viscosity.

The *ADPAC* approach can be illustrated most easily by considering the viscous flow over a flat plate as shown in Figure 3.18. A shear stress term at the wall such as:

$$\tau_{xy} = \mu \frac{\partial u}{\partial y}$$

is calculated numerically as:

$$\mu_{wall} * \frac{\Delta u}{\Delta y}$$

ADPAC Near Wall Computational Cell Structure for 2-D Flow Past a Flat Plate

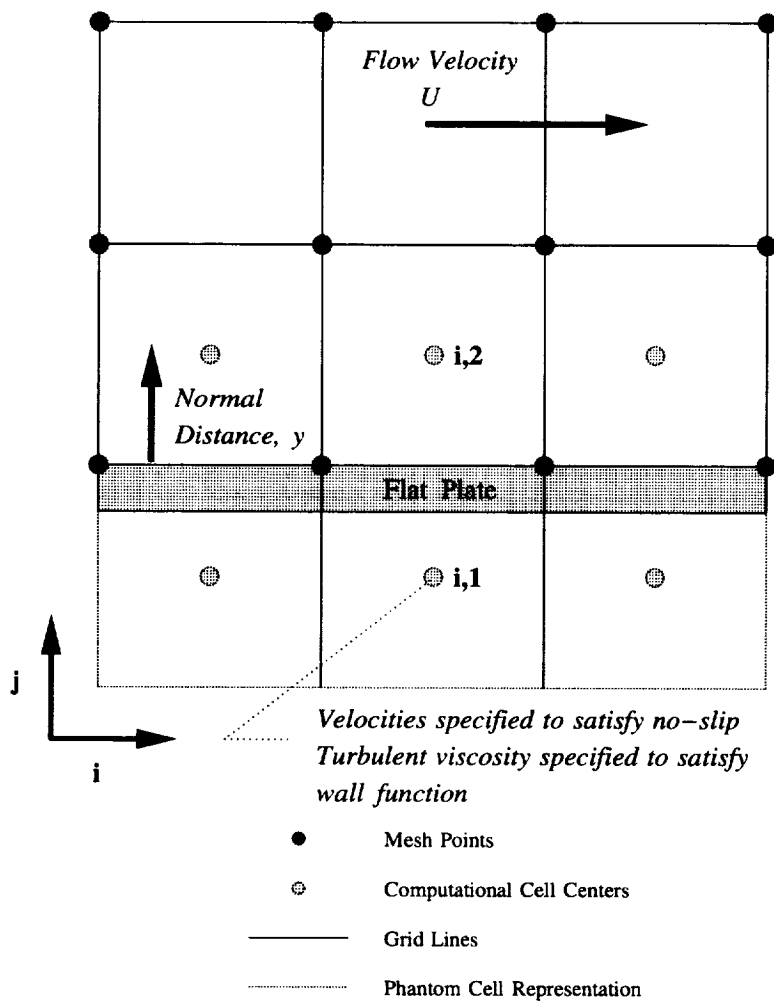


Figure 3.18: Near-wall computational structure for wall function turbulence model.

where, in this case, μ represents the combined turbulent and laminar viscosities

$$\mu_{wall} = \frac{1}{2}(\mu_{i,2} + \mu_{i,1}) \frac{(u_{i,2} - u_{i,1})}{\Delta y}$$

The subscript i, j indicates a mesh oriented cell-centered flow value, where $i, 1$ is the value in the phantom cell, and $i, 2$ is the value at the first interior cell near the wall. As mentioned, many computational schemes satisfy the shear stress requirement by modifying the wall velocity boundary condition through manipulation of the term $u_{i,1}$ in the example above. For three-dimensional flows, this becomes even more complicated as multiple velocity components must be adjusted to satisfy the overall wall shear stress. The *ADPAC* implementation instead modifies the wall turbulent viscosity boundary condition through manipulation of the single term $\mu_{i,1}$. This implies that the turbulent viscosity at the wall is non-zero, which violates the normal specification. However, since the turbulent viscosity is used exclusively in the calculation of the wall shear stress and heat conduction terms, the resulting calculations are consistent with the desired shear stress specification both in magnitude and direction, since the near-wall velocities drive the wall shear stress directly. This formulation, in effect, also implies a wall function based heat flux. For calculations in which heat transfer is unimportant, this effect is negligible. The influence of this approach on flows with heat transfer may be tested in future studies.

The shear stress specification used in the present application of wall functions is based on the following formula for the wall shear stress coefficient:

$$c_f = -0.001767 + 0.03177/Rc_n + 0.25614/Rc_n^2$$

The term Rc_n is the Reynolds number based on near-wall velocity, density, and viscosity, where the length scale is the normal distance from the wall to the first interior domain calculation cell. The wall shear stress may then be calculated from the formula:

$$\tau_{wall} = 0.5 * c_f * \rho * V_{rel}^2$$

where V_{rel} is the near-wall relative flow total velocity.

3.6.3 Two Equation Turbulence Model

Limitations associated with the implementation of the algebraic (Baldwin-Lomax) turbulence model in the *ADPAC07* code prompted the development of an advanced (two-equation) turbulence modeling capability. As part of this study, an advanced turbulence model was incorporated into the *ADPAC07* code to permit accurate prediction of a wider range of flow conditions, and hopefully improve the ability to predict highly loaded fan and compressor blade row flowfields. Initially, this effort was directed at primarily two-equation turbulence models ($k - \epsilon$, $q - \omega$, etc.) but was not necessarily intended to be limited to these models. Of particular interest was the use of “pointwise” turbulence models which do not require predetermination of the location of the nearest solid surface, as do most turbulence models. This feature provides a significant simplification of the turbulence modeling problems associated with a multiple blocked mesh code such as *ADPAC07*.

The form of the two-equation turbulence model used in the *ADPAC07* advanced turbulence model implementation was based on the two-equation $k - \mathcal{R}$ model described by Goldberg [68]. This is essentially an extension of the Baldwin-Barth [69] equation system

as implemented by Goldberg [68]. The transport equations defining this model are derived from the “standard” form of the $k - \epsilon$ turbulence model equations as follows (see e.g. Wilcox [63])

$$\begin{aligned} \frac{\partial(\bar{\rho}\tilde{k})}{\partial t} + \nabla \cdot (\bar{\rho}\vec{\tilde{V}}\tilde{k}) = \\ \nabla \cdot [(\mu + \frac{\mu_t}{\sigma_k})\nabla\tilde{k}] + P - \frac{(\bar{\rho}\tilde{k})^2}{\epsilon} \end{aligned}$$

$$\begin{aligned} \frac{\partial(\bar{\rho}\epsilon)}{\partial t} + \nabla \cdot (\bar{\rho}\vec{\tilde{V}}\epsilon) = \\ \nabla \cdot (\mu + \frac{\mu_t}{\sigma_\epsilon})\nabla\epsilon + (2 - C_{\epsilon_1})\frac{\epsilon}{\tilde{k}} - (2 - C_{\epsilon_2})\bar{\rho}\tilde{k} \end{aligned}$$

where P is the turbulent kinetic energy production term defined (for a Cartesian coordinate system) as:

$$\begin{aligned} P = \mu_t [(\nabla \tilde{u} + \frac{\partial \vec{\tilde{V}}}{\partial x}) \cdot \frac{\partial \vec{\tilde{V}}}{\partial x} + (\nabla \tilde{v} + \frac{\partial \vec{\tilde{V}}}{\partial y}) \cdot \frac{\partial \vec{\tilde{V}}}{\partial y} + (\nabla \tilde{w} + \frac{\partial \vec{\tilde{V}}}{\partial z}) \cdot \frac{\partial \vec{\tilde{V}}}{\partial z} - \frac{2}{3}(\nabla \cdot \vec{\tilde{V}})^2] \\ - \frac{2}{3}\bar{\rho}\tilde{k}(\nabla \cdot \vec{\tilde{V}}) \end{aligned}$$

By defining a new variable \mathcal{R} as

$$\mathcal{R} = \frac{k^2}{\epsilon} \quad (3.128)$$

and noting the identity

$$\frac{D\mathcal{R}}{\mathcal{R}} = \frac{2Dk}{k} - \frac{D\epsilon}{\epsilon}. \quad (3.129)$$

Baldwin and Barth subsequently developed a transport equation for \mathcal{R} from the k and ϵ equations. The final form of the $k - \mathcal{R}$ equation system can be expressed as:

$$\begin{aligned} \frac{\partial(\bar{\rho}\tilde{k})}{\partial t} + \nabla \cdot (\bar{\rho}\vec{\tilde{V}}\tilde{k}) = \\ \nabla \cdot [(\mu + \frac{\mu_t}{\sigma_k})\nabla\tilde{k}] + P - \frac{(\bar{\rho}\tilde{k})^2}{\bar{\rho}\mathcal{R}} \end{aligned}$$

$$\begin{aligned} \frac{\partial(\bar{\rho}\mathcal{R})}{\partial t} + \nabla \cdot (\bar{\rho}\vec{\tilde{V}}\mathcal{R}) = \\ (\mu + \frac{\mu_t}{\sigma_\epsilon})\nabla^2\mathcal{R} - \frac{\rho}{\sigma_\epsilon}\nabla\nu_t \cdot \nabla\mathcal{R} + (2 - C_{\epsilon_1})\frac{\mathcal{R}P}{\tilde{k}} - (2 - C_{\epsilon_2})\bar{\rho}\tilde{k} \end{aligned}$$

where:

$$\mathcal{R} = \tilde{k}^2/\epsilon$$

In developing the diffusion terms in the \mathcal{R} equation certain terms were omitted based on order of magnitude considerations (although the actual steps used in the derivation are not obvious from the authors’ description in Reference [69]).

Goldberg subsequently developed a “pointwise” turbulence model based on the $k - \mathcal{R}$ equation system based on early work by Launder and Sharma [70]. Here the adjective “pointwise” implies that the calculation of the turbulent viscosity is only dependent on flow data which is local to any point in the flow. Traditionally, most turbulence models require a calculation to determine the physical distance to the nearest wall or shear layer centerline. Since the *ADPAC07* code possesses arbitrary numbers of mesh blocks, wall boundaries, etc., determining the distance from any given mesh point to the nearest wall is a formidable computational task, and therefore, the pointwise turbulence model provides an enormous simplification.

The calculation (as reported by Goldberg [68]) of the turbulent viscosity proceeds as follows:

$$\begin{aligned}\mu_t &= C'_\mu f_\mu \rho \mathcal{R} \\ C'_{\epsilon_1} &= 1.44 \\ C'_{\epsilon_2} &= 1.92 \\ \sigma_k &= 1.0 \\ \sigma_\epsilon &= 1.3 \\ C'_\mu &= 0.09 \\ f_\mu &= f_\mu(\mathcal{R}_t) = \frac{1 - e^{-A_\mu \mathcal{R}_t^2}}{1 - e^{-A_\epsilon \mathcal{R}_t^2}} \\ A_\mu &= 2.5 \times 10^{-6} \\ A_\epsilon &= 0.2 \\ \mathcal{R}_t &= \frac{\rho \mathcal{R}}{\mu} = \frac{k^2}{\nu \epsilon}\end{aligned}$$

where \mathcal{R}_t is, in effect, the turbulence Reynolds Number.

The form of the $k - \mathcal{R}$ model is similar to other two-equation models, and the code was constructed so that it can be rapidly altered to solve other two-equation models ($k - \epsilon$, $q - \omega$, etc.) as needed. The $k - \mathcal{R}$ model is particularly attractive due to the simplified solid surface boundary conditions.

$$k = 0 \quad \mathcal{R} = 0 \tag{3.130}$$

and flowfield initialization

$$k \approx 1 \times 10^{-5} \quad \mathcal{R} \approx 1 \times 10^{-6} \tag{3.131}$$

Numerical implementation of the two-equation turbulence modeling strategy involves several differences from the solution procedure described for the Reynolds-averaged Navier-Stokes equations. The $k - \mathcal{R}$ equations were advanced in time using the same finite volume discretization and Runge-Kutta time marching procedures described in previous sections. No added dissipation was used for the $k - \mathcal{R}$ equations. Instead, the convective cell face flux evaluations were performed using a first order upwind approximation for the flow variables on the cell face. The k and \mathcal{R} variables themselves were limited by enforcing the conditions $k, \mathcal{R} \geq 1 \times 10^{-6}$. The production term P was also required to have a nonnegative value. Validation studies illustrating the quality of solutions obtained with the *ADPAC07* $k - \mathcal{R}$ turbulence model are given in Chapter 6.

3.7 Overall *ADPAC07* Numerical Solution Procedure

The overall solution procedure begins by defining a set of initial data, and advancing the solution from that point forward in time until the desired solution (steady state, time-periodic, or finite time interval) has been reached. Initial data is normally specified as a uniform flow, or may be read in as a “restart” of a previous existing solution. Normally, for steady flow calculations, the “full” multigrid startup procedure is utilized to accelerate convergence by initializing the solution on a coarse mesh before incurring the expense of fine mesh iterations. Steady state solutions are normally deemed converged when the average residual R has been reduced by a factor of 10^{-3} , or when the residual has ceased to be reduced. Experience has shown that pressure-driven flow quantities generally converge first (e.g. mass flow, lift, etc.) while viscous driven flow quantities converge after more iterations (e.g. loss). In many cases, the average residuals may appear to be converged, while integrated quantities such as loss continue to change. Solution convergence must also be interpreted with this behavior in mind. It is possible that for some steady flow calculations, the solution is truly unsteady (i.e. - vortex shedding behind a circular cylinder) and in these cases the residual may not be reduced beyond a certain limit.

Chapter 4

Code Parallelization

4.1 Introduction

One of the practical difficulties of performing CFD analyses is finding sufficient computational resources to allow for adequate modeling of complex geometries. Oftentimes, workstations are not large enough, and supercomputers have either long queues, high costs, or both. Clearly, a means of circumventing these difficulties without giving up the flexibility of the CFD code or the complexity of the model would be welcome. One possibility is to write a code which could run in parallel across a number of processors, with each one having only a piece of the problem. Then, a number of lesser machines could be harnessed together to make a virtual supercomputer.

The most likely candidates for creating such a machine are desktop workstations which are fully loaded during the day, but sit idle at night. Tremendous computational power could be made available at (ideally) no extra hardware investment. There are also massively parallel computers available on the market designed specifically for such applications. These machines are aiming at order of magnitude improvements over present supercomputers.

The problem of course, lies in the software. Parallelization is today about as painful as vectorization was a decade ago. There is no standard parallel syntax, and no compiler exists which can automatically and effectively parallelize a code. It is difficult to write a parallel code which is platform independent. What makes things worse is that there is no clear leader in the parallel computer industry, as there has been in the supercomputer industry.

The objective behind the development of the consolidated *ADPAC07* described in this paper was to create a platform independent parallel code. The intent was to design a parallel code which looks and feels like a traditional code, capable of running on networks of workstations, on massively parallel computers, or on the traditional supercomputer. User effort was to be minimized by creating simple procedures to migrate a serial problem into the parallel environment and back again.

4.2 Parallelization Strategy

The *ADPAC07* code has some innate advantages for parallelization: it is an explicit, multi-block solver with a very flexible implementation of the boundary conditions. This presents two viable options for parallelization: parallelize the internal solver (the “fine-grained” approach), or parallelize only the boundary conditions (the “coarse-grained” approach). The fine-grained approach has the advantage that block size is not limited by processor size.

This is the approach frequently taken when writing code for massively parallel computers, which are typically made up of many small processors. The coarse-grained approach is favored when writing code for clusters of workstations, or other machines with a few large processors. The dilemma is that a parallel *ADPAC07* needs to run well on both kinds of machines.

The fine grained approach is especially enticing for explicit solvers. Explicit codes have proven to be the easiest to parallelize because there is little data dependency between points. For a single block explicit solver, fine-grained parallelization is the clear choice. However, with a multiblock solver, the boundary conditions must be parallelized in addition to the interior point solver, and that can add a lot of programming effort. The coarse-grained approach is admittedly easier for multi-block solvers, but what if the blocks are too big for the processors? The simplest answer is to require the user to block out the problem so that it fits on the chosen machine. This satisfies the programmer, but the user is faced with a tedious chore. If the user decides to run on a different machine, then the job may have to be redone. The pain saved by the programmer is passed directly to the user.

A compromise position was reached for the parallel *ADPAC07* code. The coarse-grained approach is used, but supplemental tools are provided to automatically generate new grid blocks and boundary conditions for a user-specified topography. In this way, the parallel portions of the code are isolated to a few routines within *ADPAC07*, and the user is not unduly burdened with architecture considerations. Details of running the *ADPAC07* in parallel are given in a later chapter.

4.3 Description of the *ADPAC07* Parallel Implementation

This section describes the basic structure of the consolidated *ADPAC07* code, a 3-D Euler/Navier-Stokes analysis developed for the express purpose of providing a robust, flexible aerodynamic analysis tool for aerospace propulsion applications. The analysis is capable of predicting both steady state and time-dependent flowfields, and was structured to be capable of either serial execution or parallel execution on massively parallel or workstation cluster computing platforms from a single source code. The serial/parallel execution capability is determined solely at compilation by a simple library substitution.

Researchers at IUPUI¹ parallelized an early version of *ADPAC07* using the Application Portable Parallel Library (APPL) [76] communications library. This code was ported to an nCUBE 2 massively parallel computer at Allison by porting the necessary APPL routines to the nCUBE. The APPL library was developed at NASA, and is designed to allow applications to run in parallel on many architectures without changing the source code. It is essentially an interface between the application and the parallel computer (or cluster of workstations) which insulates the application from machine dependent functions. At present APPL is available for Intel iPSC/860, Intel Delta, Alliant, Silicon Graphics workstations, and IBM RS6000 workstations and workstation clusters. Since each machine has different message passing protocols (UNIX sockets, node-to-node communications, etc.), there are compiler directives which select the appropriate routines for a particular machine.

The APPL library includes routines for sending and receiving messages, and a number of routines for timing, processor identification, and checking the message buffers. Each of these routines has a straightforward analog in nCUBE system routines, which were added

¹IUPUI is Indiana University/Purdue University at Indianapolis

to APPL with a compiler flag to identify them. APPL also has other features which are not required by *ADPAC07*, which were not ported as part of this task.

Having ported the necessary APPL routines to the nCUBE, the IUPUI version of *ADPAC07* ran with virtually no changes on the nCUBE. Two small test cases were run, one with one block per processor, and one with multiple blocks per processor. Both converged to the correct answer on various numbers of processors. However, there were problems when a larger problem was run on a large number of processors. The execution aborted because of messages overflowing the communications buffer. The nCUBE at Allison has 64 processors, each with 4 megabytes of memory. For each job, memory is allocated for the executable image, data storage, communications buffer, stack and heap buffers on each processor. It was determined by running other problems, that an inordinate amount of buffer space is used by the code.

The reason for the large buffer requirements is that all processors send data to each other simultaneously, and then all processors read their incoming messages. In this scheme, the communications buffer holds all of the boundary data at one time. This also means that temporary storage must be allocated to hold all of the incoming data until the boundary condition routines can use it. Actually, each flow variable is sent and received separately, reducing the requirements by a factor of six. This still leaves a communications requirement equal to 1/3 or more of the data on each processor.

An alternative communications scheme takes advantage of the way boundary conditions are applied in *ADPAC07*. Boundary conditions are applied by looping over an ordered list of conditions, with the appropriate blocks participating at each step. In parallel, each processor loops over the same ordered list, but participates only in the conditions which require action by one of the blocks assigned to it. If a processor is not involved in a particular boundary condition, it continues to loop through the boundary condition list. Since each boundary condition involves, at most, two blocks (and therefore, at most two processors), this provides a natural mechanism for scheduling communications efficiently.

For each boundary condition in the ordered list, each processor checks to see if it needs to send data to another processor. If so, it sends a message to the appropriate processor indicating that it is ready to send. It then waits for acknowledgement from the receiver before sending the required data. If the boundary condition does not call for the processor to send data, it then checks to see if it needs to receive data. If so, it waits until the ready to send message comes from the appropriate processor, and then returns an acknowledgement to begin the communications. In this way, no more than one data packet is in the communications buffer at any time.

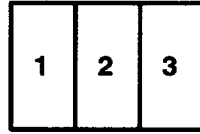
While there is a performance penalty associated with processors waiting to synchronize communications, there is a potential performance improvement in removing the communications bottleneck which occurs when all processors send and receive simultaneously. The boundary conditions are still applied in parallel, because each processor waits only when it is involved in communications for a particular boundary condition. There is no delay for boundary conditions which do not require interprocessor communications. Figure 4.1 illustrates the differences between the two communication schemes.

Since communications are performed only when needed, the communications procedures need to be coded into the boundary conditions. At present, there are relatively few boundary conditions which could potentially require interprocessor communications. The most common of these is the **PATCH** condition (see [28]), which requires the flow variables from a neighboring face. To accommodate parallel computations without disturbing the original patching algorithm, a subroutine was added to load small temporary arrays with the

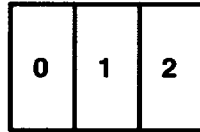
ADPAC Parallel Communication

Sample Problem

**Physical
Blocks**



**Processor
Map**



Boundary Conditions

B. C.	Sender	Receiver
1. Patch 1 2	Proc 1	Proc 0
2. Patch 2 1	Proc 0	Proc 1
3. Patch 2 3	Proc 2	Proc 1
4. Patch 3 2	Proc 1	Proc 2

Original Procedure	Revised Procedure																																
<div>1. Initialize problem</div> <div>2. Solve interior points</div> <div>3. All processors send data</div> <div>4. All processors receive data</div> <div>5. Compute B.C.'s</div> <div><table><tr><td>Proc:</td><td>0</td><td>1</td><td>2</td></tr><tr><td>a.</td><td>BC 1</td><td>BC 2</td><td>BC 3</td></tr><tr><td>b.</td><td></td><td>BC 4</td><td></td></tr></table></div> <div>6. Repeat steps 2 – 5</div>	Proc:	0	1	2	a.	BC 1	BC 2	BC 3	b.		BC 4		<div>1. Initialize problem</div> <div>2. Solve interior points</div> <div>3. Compute B.C.'s</div> <div><table><tr><td>Proc:</td><td>0</td><td>1</td><td>2</td></tr><tr><td>a. BC 1</td><td>rcv</td><td>snd</td><td></td></tr><tr><td>b. BC 2</td><td>snd</td><td>rcv</td><td></td></tr><tr><td>c. BC 3</td><td></td><td>rcv</td><td>snd</td></tr><tr><td>d. BC 4</td><td></td><td>snd</td><td>rcv</td></tr></table></div> <div>4. Repeat steps 2 – 3</div>	Proc:	0	1	2	a. BC 1	rcv	snd		b. BC 2	snd	rcv		c. BC 3		rcv	snd	d. BC 4		snd	rcv
Proc:	0	1	2																														
a.	BC 1	BC 2	BC 3																														
b.		BC 4																															
Proc:	0	1	2																														
a. BC 1	rcv	snd																															
b. BC 2	snd	rcv																															
c. BC 3		rcv	snd																														
d. BC 4		snd	rcv																														
<div>Notes:</div> <div>Original procedure requires very large communications buffer, but communications are isolated in the code.</div>	<div>Revised procedure requires small buffer, but communications must be coded into boundary conditions. Synchronous communications insures scalability to more processors. Memory requirements are reduced because fewer temp arrays are needed for incoming data.</div>																																

Figure 4.1: Improved communication scheme reduces memory requirements.

required data, performing communications as necessary. The original patching subroutine is then called with the temporary arrays passed as arguments. Other boundary conditions, such as the mixing plane routine and the radial equilibrium routine are handled in a similar manner.

4.4 Domain Decomposition

One of the most difficult problems in parallel computing is determining how to subdivide the original problem into a balanced set of smaller pieces. Since the solution is paced by the slowest processor, there is a significant speed penalty associated with load imbalance.

The IUPUI version of *ADPAC07* was capable of automatically subdividing the solution domain, according to user specifications. The code determines what additional boundary conditions are required and this involves specifying the number of divisions in each indicial direction for each block in a file named *casenamc.parallel*. This adds very little work to run an *ADPAC07* case in parallel.

Using this approach, each processor reads in the entire block containing the piece assigned to it, and then runs only its piece, resulting in a massive waste of memory. For example, for a single block problem, only $1/N$ of the memory is actually used on a case running with N processors. This is obviously unacceptable for large numbers of processors.

There are two obvious solutions to this problem. The user could create the *ADPAC07* input with the appropriate number of blocks for parallel computation initially, so that no further subdivisions are required. This could be quite time consuming, and would be particularly unpleasant if a restart was desired on a different number of processors.

The other obvious solution is to use the subdivision algorithm in the parallel *ADPAC07* to do the decomposition and write out new input files. A parallel preparation program was created (*SIXPAC*) from the IUPUI version of *ADPAC07* which accomplishes this task. This program, running on a single processor, reads an *ADPAC07* input set and writes out the new subdivided grid file and the appropriate boundary condition file. The flow solution is then computed in parallel as a separate job, which writes out restart files in the subdivided form. A program to reassemble the pieces into a single restart file compatible with the original input was also developed (*BACPAC*). In this way, a job can be run on any number of processors, and then postprocessed in the same way as the serial version of *ADPAC07*. A job could also be restarted on a different number of processors by rerunning the preparation code with new subdivisions.

In any event, the burden of insuring balanced blocks remains with the user. Experience with other codes running on the nCUBE has shown that simply dividing the domain with roughly equal numbers of points on each processor is satisfactory. When running multigrid, some care must be exercised to insure that multigrid numbers are preserved in the new blocks.

4.5 *SIXPAC* (Block Subdivision) Program

SIXPAC, which stands for Subdivision and Information eXchange for Parallel Adpac Calculations, enables the user to redefine the block structure of an *ADPAC07* job. Using *SIXPAC*, large grid blocks can be subdivided to improve load balance, or to make use of smaller memory processors in parallel calculations. *SIXPAC* generates new input, mesh, restart, and boundata files for the subdivided problem, creating new blocks according to

user specifications. The resulting files represent a problem equivalent to the original, but with more, smaller, blocks. Although the number of unique grid points is unchanged, the total number of points is larger because of duplication at interfaces.

The motivation for *SIXPAC* comes from the way *ADPAC07* was parallelized. Rather than parallelize the interior point solver, *ADPAC07* was parallelized through the boundary conditions. An individual block cannot be run across multiple processors; each processor must contain only whole blocks. This implies that a problem with a single large block couldn't be run in parallel. *SIXPAC* enables large blocks to be recast as groups of smaller blocks, so that they can be run in parallel. *SIXPAC* is not required to run a problem in parallel, but it simplifies the process of setting up a problem for optimal parallel performance.

The preprocessor reads the original large blocks, subdivides them for the desired number of processors, and generates appropriate boundary conditions based on the original boundary condition list and on the subdivision scheme. Generating boundary conditions for test cases uncovered some errors and omissions in the boundary condition generation routine. In general, determining new boundary conditions between neighboring blocks is a complicated problem, as depicted in Figure 4.2

The complications arise from non-aligned blocks and from indices which run in opposite directions in neighboring blocks. Even when the original blocks are nicely aligned, the subdivision scheme can lead to non-aligned blocks and additional boundary conditions. The general case includes non-aligned blocks in both the "m" and "n" directions, and possibly indices which are opposed in both directions. The boundary condition generation routine was rewritten to handle all of these possibilities without user intervention.

Input files contain information which specifies how the blocks are to be subdivided. The required information includes the number of original blocks, and how each block is to be subdivided in each indicial direction (i, j, and k). In each direction, the number of subdivided blocks, and possibly the locations of the subdivisions, must be specified. If the number of subdivided blocks in a particular coordinate direction is set to 1, then the block is not divided in that coordinate direction.

By default, blocks are split into the specified number of equal sized pieces. If there is a remainder, it is spread over the processors to create nearly equal sized pieces. If unequal divisions are required in a particular direction, then the location of each division must be specified in that direction.

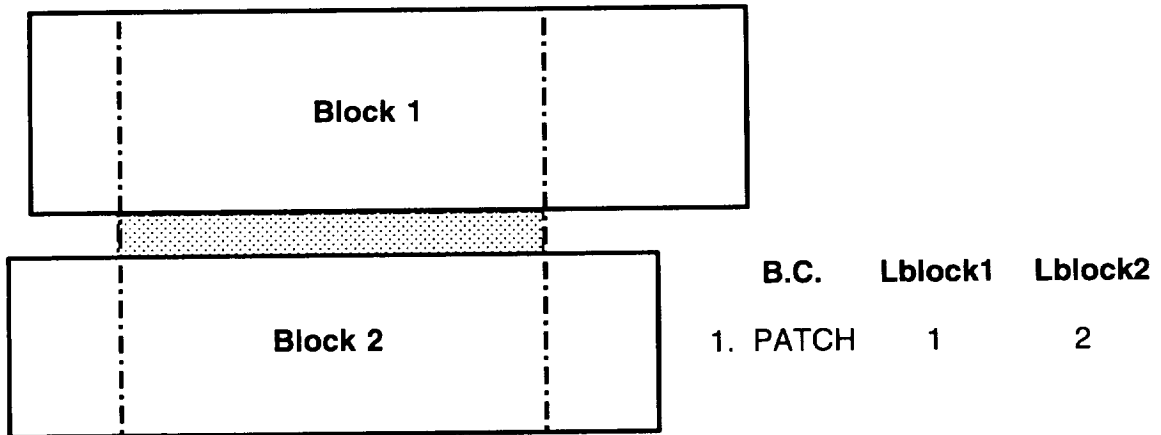
Unequal divisions are often employed to preserve levels of multigrid, or to put the edge of a geometric feature on a block boundary. Figure 4.3 illustrates how different block strategies affect multigrid. If, for example, there are 21 points in the I direction of a block, 3 levels of multigrid are possible. If this block is divided into two equal pieces of 11 points each, then only 2 levels of multigrid are possible. However, if the block is split into a block with 13 points and a block with 9 points, 3 levels of multigrid are still possible.

If a restart file is to be created for the subdivided problem, the input trigger **FREST** must be set equal to 1.0 in the *casename.input* file. This tells *SIXPAC* to look for a *casename.restart.old* file, and to subdivide it. A *Ncasename.restart.old* file is written, and the new input file will be set up to run with the new restart file.

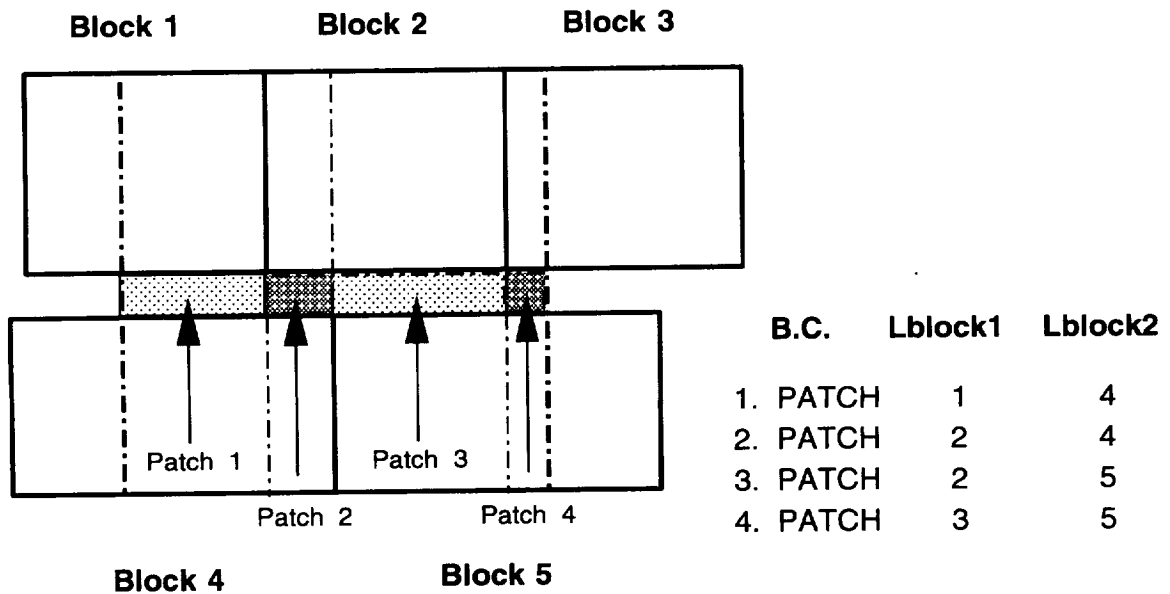
4.6 *BACPAC* Block Reconstruction Program

BACPAC, which stands for **B**lock **A**ccumulation and **C**onsolidation for **P**arallel **A**dpac

Creation of New Boundary Conditions for Subdivided Blocks



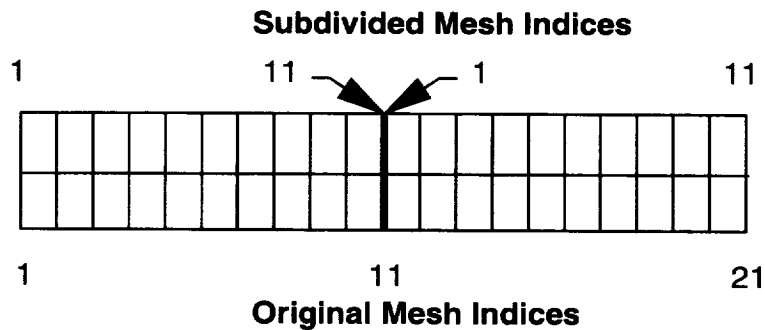
Original blocks with PATCH boundary



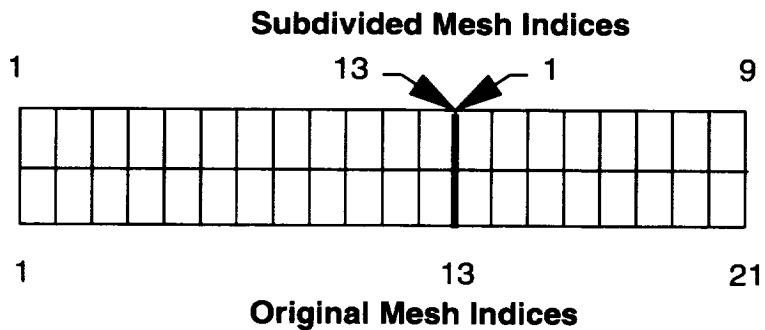
New blocks with PATCH boundaries

Figure 4.2: Boundary condition generation between neighboring blocks often results in large numbers of patches.

Subdivision of Blocks to Preserve Levels of Multigrid



Subdivision into two equal pieces results in blocks with 11 points. Only two levels of multigrid are possible, even though three levels were possible for the original block.



Subdivision into two unequal blocks, one with 13 points and one with 9 points, yields a grid capable of three levels of multigrid, like the original block.

Figure 4.3: Careful block division can preserve levels of multigrid.

Calculations, reassembles subdivided *ADPAC07* files into their original, undivided form. It is used in conjunction with *SIXPAC*, and performs essentially the inverse operation of *SIXPAC*. *BACPAC* can reconstruct mesh, PLOT3D, or restart files, producing new files which are equivalent to what would have been produced had the problem been run with the original, undivided blocks. Using *SIXPAC* and *BACPAC*, a problem can be subdivided and reconstructed any number of ways to take advantage of available computer resources.

The output files produced by *BACPAC* are the *Ncasename.msh.bac* file, the *Ncasename.p3dabs.bac* and the *Ncasename.p3drel.bac* files, and the *Ncasename.restart.bac* file. The *.bac* suffix is used to avoid confusion with existing files. Generally the *Ncasename.msh.bac* need not be created because it is identical to the original *casename.msh* file.

4.7 Load Balancing via Block/Processor Assignment

Load balancing is a critical issue for parallel computing tasks. While it is beyond the scope of this program to perform detailed load balancing analyses for every parallel computing platform tested, it seems reasonable to provide some form of control in order to distribute computational tasks efficiently across a parallel computing network. In the parallel *ADPAC07* code, this is best accomplished through manipulation of the block/processor distribution scheme. By default, the parallel operation of the *ADPAC07* code provides an automatic block to processor assignment by dividing up the blocks as evenly as possible, and, to the greatest degree possible, assigning sequential block numbers on a given processor. For example, if 8 blocks were divided between 3 processors, blocks 1, 2, and 3 would be assigned to process #0, blocks 4, 5, and 6 to processor #1, and blocks 7, and 8 to processor #2 (note that the processor numbering scheme is 0, 1, 2, etc.). This procedure is nearly optimal when each block is the same size, and each processor has the same computational power. Unfortunately, our experience is that block sizes and computational resources often vary dramatically. In this regard, a system was developed which permits the user to specify the block to processor assignment through a special input file (*casename.blkproc*). A sample *casename.blkproc* file is given below for an 8 block mesh distributed across 4 processors:

```
number of blocks
8
block #      proc #
1           0
2           1
3           1
4           1
5           2
6           2
7           2
8           3
```

In the case described by the above file, block 1 is assigned to processor #0, blocks 2, 3, and 4 to processor #1, blocks 5, 6, and 7 to processor #2, and block 8 to processor #3. This block assignment might be advisable for the case when blocks 1 and 8 are significantly larger in size than the other blocks, or if processor #0 and #3 have less memory or a slower CPU than the remaining processors. The default block assignment scheme previously described is employed when the *casename.blkproc* file is not defined.

4.8 Serial/Parallel Solution Sequence

In order to run *ADPAC07* in parallel, *ADPAC07* must be compiled for parallel execution. The chapter on code compilation in the *ADPAC07* User's Manual [28] describes the proper compilation procedure.

ADPAC07 is parallelized using the Application Portable Parallel Library (APPL) message passing library, developed at NASA Lewis. Reference [76] explains how to write code using APPL and how to run codes written with APPL. While APPL runs on many platforms, this section will deal with only two of them: homogeneous workstation clusters and nCUBE massively parallel computers. These two platforms are representative of how *ADPAC07* runs in parallel. The APPL document should be consulted for cases not covered in this manual.

Regardless of the platform, running *ADPAC07* in parallel requires the APPL **compute** function and a *procdef* file. Codes running under APPL are not initiated by typing the executable name, but use the APPL **compute** function instead. The syntax for executing *ADPAC07* is as follows:

compute < *cascname.input* > *output*

The **compute** function controls the execution of *ADPAC07* on the various processors, taking additional input from the *procdef* file. The *procdef* file contains the names of the executable images and the processors that they are to be loaded on. The **compute** function establishes communications with each processor specified in the *procdef* file, loads the *ADPAC07* executable image, and initiates the run on each processor. Also, the **compute** function oversees the running processes, monitoring the processors for abnormal terminations. If a communications error is trapped, or if a process has died unexpectedly, the **compute** function shuts down all of the remaining processes gracefully. This feature is most important on workstation clusters, which have no built-in mechanism for monitoring parallel jobs.

The normal *ADPAC07* input file is redirected from standard input to the **compute** function. The redirected input is available to all of the processes (although *ADPAC07* currently does all reading from node 0). The output file may be redirected, or allowed to stream to the terminal, just as in a serial execution.

The *procdef* file should appear in the directory where the job is being run. It has a different syntax for the various parallel platforms. The simplest formulation is for hypercube machines (nCUBE and Intel). A sample *procdef* file for an nCUBE 2 is as follows:

```
someuser frntend . ./adpacp.ncube -1 4
```

The first token in the *procdef* file is the user name (someuser). The second token is the name of the front-end processor to the nCUBE 2. The third token is the path to the directory for input and output files (in this case, the current directory, "." is used). The fourth token is the executable name (the path may be specified to be sure the correct executable is used). The fifth token specifies how the processors are mapped (-1 indicates hypercube ordering, -2 indicates mapping into a ring). Hypercube ordering is generally preferred. The last token specifies the number of processors to be allocated (4 in this case).

Similarly, a sample *procdef* file for a workstation cluster is as follows:

```
someuser host1 . 1 adpacp.aix
someuser host2 . 1 adpacp.aix
someuser host3 . 2 adpacp.aix adpacp.aix
```

In this example, the first three tokens represent the user name, host name and the path to the working directory, just as before. The fourth token indicates the number of processes to be run on the host, and the remaining tokens are the executable images corresponding to the processes. The last line of the example shows 2 processes running on host3. When more than one process is specified for a given host, an equivalent number of executable image names *must* be specified. Using this *procdef* file, the virtual parallel computer will consist of four processes running on three workstations.

The host machines in a workstation cluster must be connected by ethernet, but do not have to share disks, or be part of the same subnet. This provides tremendous flexibility in constructing a workstation cluster. However, the primary performance bottleneck encountered on workstation clusters involve the network. The benefits of adding processors may be more than offset by poor network performance. The tradeoff varies with the problem and with the hardware configuration.

In general, the behavior of *ADPAC07* in parallel is the same as in serial. This is especially true if there are no input errors. The output files may be different if there are input errors. There are two general types of input errors detected in *ADPAC07*. Errors involving the grid or the input file will generally be detected by all processors, and the error messages will appear as they do in serial.

If, however an error is discovered in a boundary condition routine, the output messages will probably appear differently in the output file, and may not appear at all. Since *ADPAC07* boundary conditions are applied in parallel, node 0 does not execute all of them, but only those involving a block assigned to node 0. If node 0 does not encounter the error, then a different node writes the error message. Since the writing node is out of sync with node 0, the error message may be written to a different place in the output file than if node 0 had written it.

Buffering of output on the various processors can also cause a problem. Usually, after an error message is printed, execution is stopped on all processors. If execution is stopped before the buffer is flushed, then output may be lost from some processors. The result is that an error message could be caught in the buffer and never appear in the output file (not all systems flush the buffer conveniently when a process “hangs”). If *ADPAC07* terminates for no apparent reason, this may explain the problem. The solution is to rerun the job without redirecting the output. If output is not redirected, it is normally not buffered, and all of the output will appear.

It is also possible to get multiple copies of an error message if more than one processor encounters the error. Wherever possible, *ADPAC07* has been coded to avoid these problems, but these unfortunate possibilities still exist. Therefore, running *ADPAC07* interactively is the best way to track down input problems.

Aside from these considerations, running *ADPAC07* in parallel is very much like running *ADPAC07* in serial. The input files are identical, and the output files are very similar. The most common problems in running *ADPAC07* in parallel are failing to use the **compute** function, improperly specifying the parallel configuration in the *procdef* file, and attempting to run a serial executable in parallel.

4.9 Performance of the *ADPAC07* code in Parallel

Code performance data was generated for parallel *ADPAC07* by successively doubling and successively halving a problem on an nCUBE 2 computer. Limited data was also collected

from runs on the LACE (Lewis Advanced Cluster Environment) cluster of IBM RS-6000 workstations, but these results are not as meaningful because the runs were not made in a dedicated environment. Speed factors were computed by dividing the time for a single processor run by the times for multiple processor runs.

The problem chosen for the nCUBE was a subsonic duct flow with an initial $17 \times 17 \times 17$ grid. This is computationally similar to a cascade flow, but is much simpler to manipulate when halving and doubling. All runs were made for 50 iterations without multigrid. The turbulence model was activated at the 10th iteration.

Timing data was collected for four categories: time in initialization, time computing the solution, time writing output files, and time spent applying boundary conditions. CPU time from node 0 was collected for each category, which is equivalent to the wall clock time in the dedicated nCUBE environment. The initialization and output times may be skewed by contention with others for the disk drive, which was not dedicated. The disk drive was nfs mounted to the nCUBE front end, so network traffic could also affect I/O (input/output) times. The solution and boundary condition times are not affected by either of these situations.

The purpose of successively doubling the problem is to determine scalability to larger problems running on more processors. The successive halving shows the speed benefit of running the same problem on more processors. In successive doubling, the same problem was rerun with the grid size doubled in each direction. The number of processors was also doubled in each direction, so every processor has the same ($17 \times 17 \times 17$) number of grid points. The blocks were patched together to form a single solution domain.

In successive halving, the same problem was rerun on more processors with proportionately smaller grids. The grids are not exactly half the size of their parents because the edge points of each block must be repeated to insure mating block faces. This becomes significant as the number of processors grows. The single processor case uses a $17 \times 17 \times 17$ block for a total of 4913 points, but the 64 block case has 65 $5 \times 5 \times 5$ blocks for a total of 8000 points. If the intent is to determine the speedup from further subdivision of a problem, then a comparison of raw solution times would be appropriate. If, however, the intent is to determine the scalability of the code, normalizing the solution times by the total number of grid points would be in order.

nCUBE2 Results

Figure 4.4 shows the raw data from the successive halving series of runs. An alarming trend is noted in the total times: the times drop and the rise again as the number of processors is increased. As seen in the figure, the principal cause of this is a significant increase in the I/O time (initialization plus output times).

Further examination shows that the input time is substantially worse for larger numbers of processors. There are two possible explanations for this: either the broadcasting of the data is slow, or the parsing of the input and boundary data files is slow. The parsing routine does many character comparisons for each input and boundary item. These seem particularly slow when watching the output file stream by. Since the boundary data file sizes roughly correlate with the initialization curve, this is likely to be the largest contributor. However, the correlation is not exact, indicating that reading and broadcasting the grid is also significant. The shape of the initialization curve is pretty much as expected since the grid and boundary data files are growing with each successive halving.

The output times are roughly constant with problem size, which is probably evidence of network and disk contentions since the amount of output grows with problem size. Since

***Parallel ADPAC Timing Runs
Successive Halving of Problem Size***

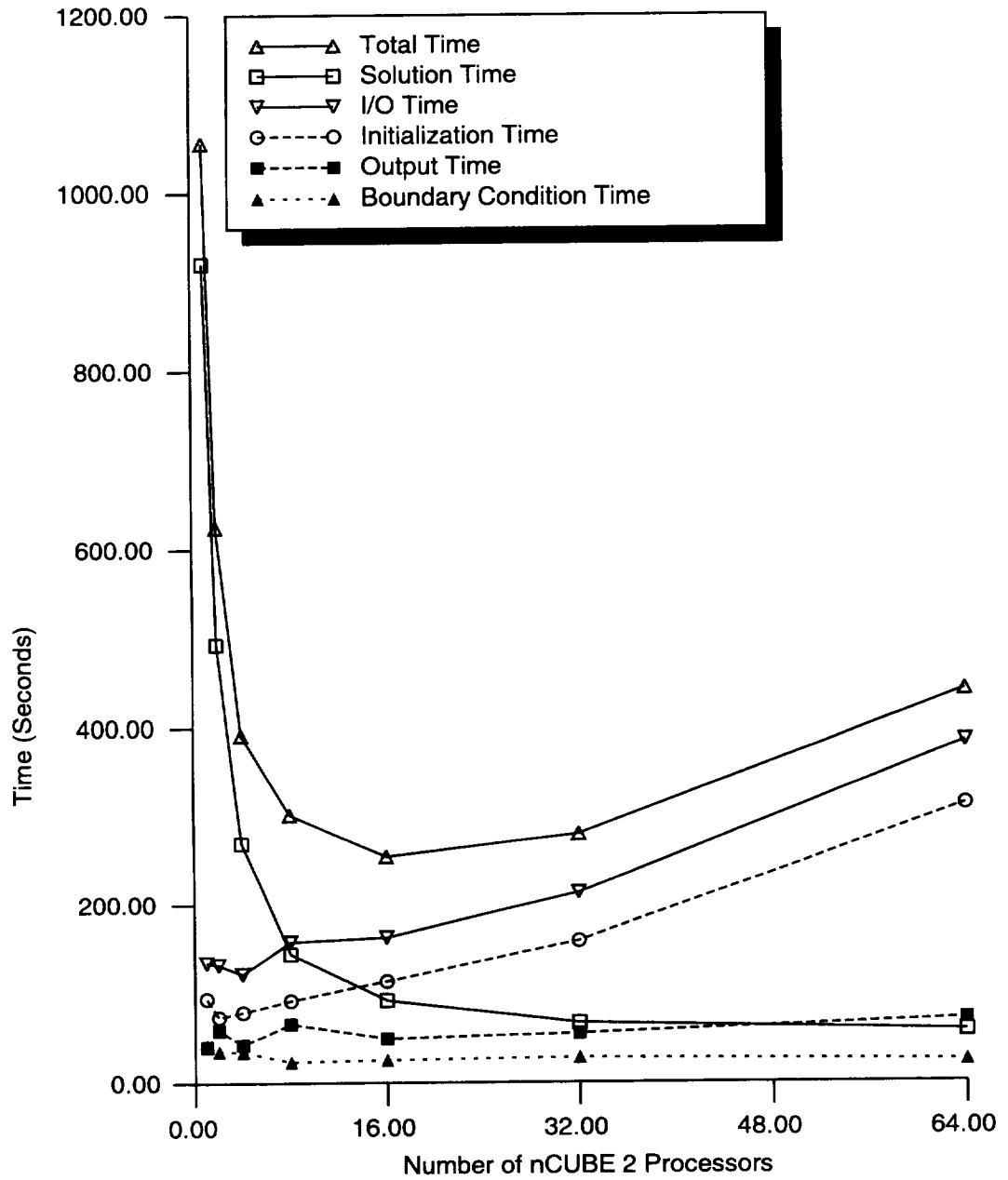


Figure 4.4: Comparison of CPU times for a problem subdivided over a range of processors.

the output time is small, it is difficult to assign much significance to its behavior.

In any case, the I/O times are a fixed cost of making a parallel run: it takes the same amount of time regardless of the number of iterations. For airfoil cascades, 200 iterations are generally enough to reach convergence on a fine mesh using three levels of multigrid. The significance of the I/O time is therefore reduced by at least a factor of 4 in practical problems.

One surprising result is that the boundary condition times decrease with more processors. Since each boundary condition involves fewer points as the number of processors grow, there is a downward trend on the time required, but the communications requirements grow substantially. This indicates that the communications overhead is not important for patched blocks on the nCUBE.

The solution time decreases uniformly with each successive halving, as expected. The slope of the curve decreases because more points are required for each successive run.

Figure 4.5 shows the speed factors achieved in the successive halving sequence of runs. Ideally, the speed factor would be the number of processors, but actual performance is degraded for many reasons.

The curve labeled "Soln time" shows the speed factors based on raw CPU times for the solver in each run. The curve labeled "Soln Time/Grid Point" shows the speed factors on a normalized basis.

In either case, the numbers are not very impressive. The principal cause of the degradation is that the surface/volume ratio is increasing as the number of processors grows. For example, the single block case has $6 \times 17 \times 17$ or 1734 surface points with 4913 points in the volume, while the 64 block case has $6 \times 5 \times 5$ or 150 surface points with only 125 points in the volume. Thus, the relative importance of the boundary condition time grows with the number of processors. The lesson here is that code performance is best when each node is fully loaded. Using more processors to solve the same problem will reduce solution time, but with high efficiency.

Figure 4.6 shows the speed factors resulting from successive doubling of the problem. In this series, three runs were made: 1 processor, 8 processors, and 64 processors, each using a $17 \times 17 \times 17$ grid. As seen, the speed factors are quite good, indicating that parallel *ADPAC07* can be effectively applied to large problems. The parallel effectiveness of the code is 89.4% using 64 processors. Actual performance will vary with load distribution, grid size, and the type of boundary conditions used. Something on the order of 50% more nodes will fit on each nCUBE processor, so performance could be slightly better, but the grid size was chosen for convenience in successive halving.

Figure 4.7 shows the raw CPU times for successive doubling of the problem. Again, the total times are not terribly impressive, but for different reasons than for the halving case. In this figure, the solution times are excellent: only a 12% increase in overall solution time when the problem size increased by a factor of 64 (number of processors also increased by a factor of 64). The growth in the boundary condition time accounts for this increase. The total times look poor because of the output time. The slope of the output time curve shows a nearly linear growth with problem size, which is expected.

The initialization time does not show the same drastic increase as the output time. The initialization time is comprised of input processing, boundary data processing, and reading the grid. It was seen above that the parsing of character data is slow on the nCUBE. Additional timing data is needed to explain the initialization performance. The output files are 15.3 times bigger than the grid file, so I/O performance problems will be accentuated in the output routines.

***Parallel ADPAC Timing Runs
Successive Halving of Problem Size***

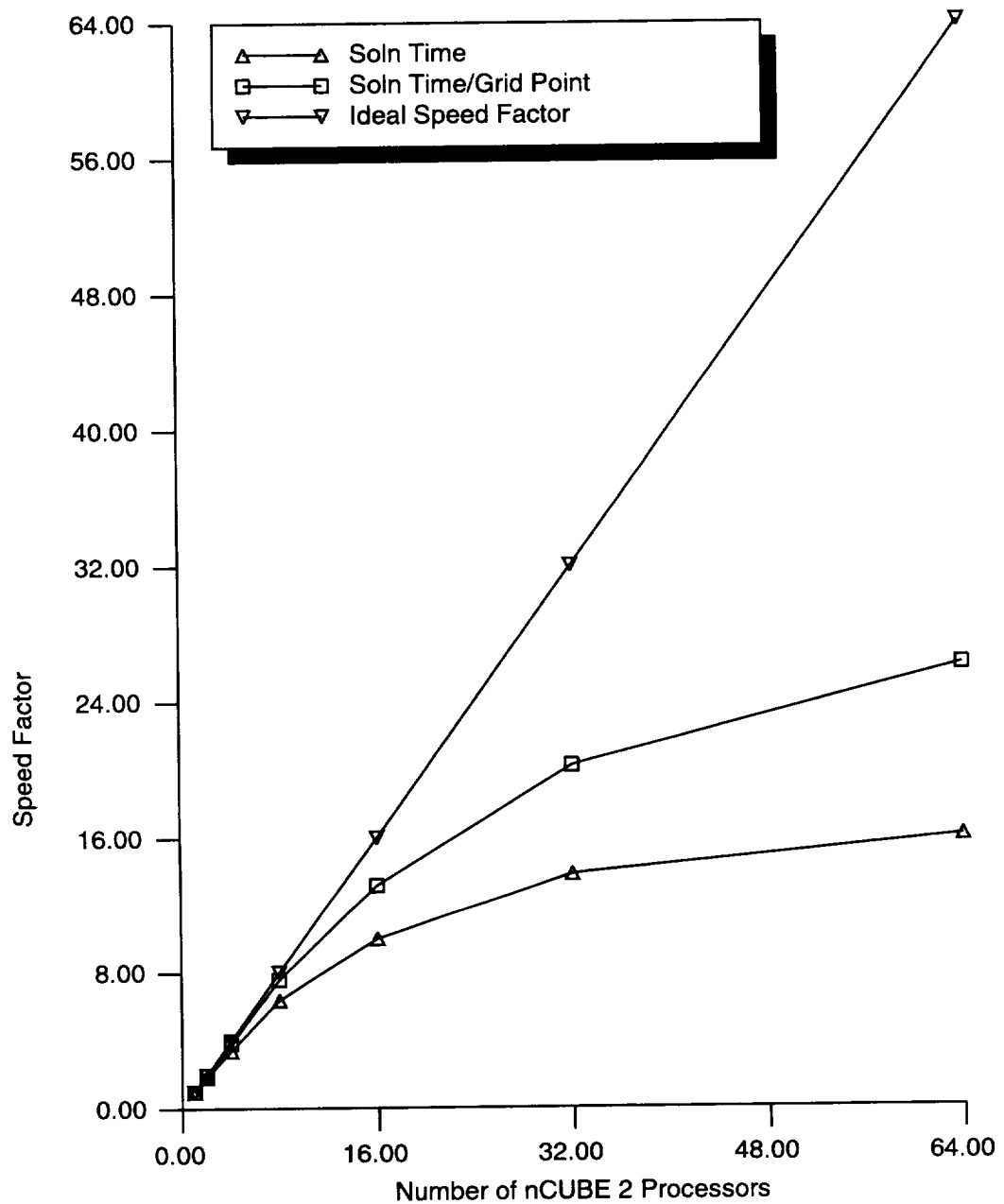


Figure 4.5: Comparison of speed factors for a problem subdivided over a range of processors.

***Parallel ADPAC Timing Runs
Successive Doubling of Problem Size***

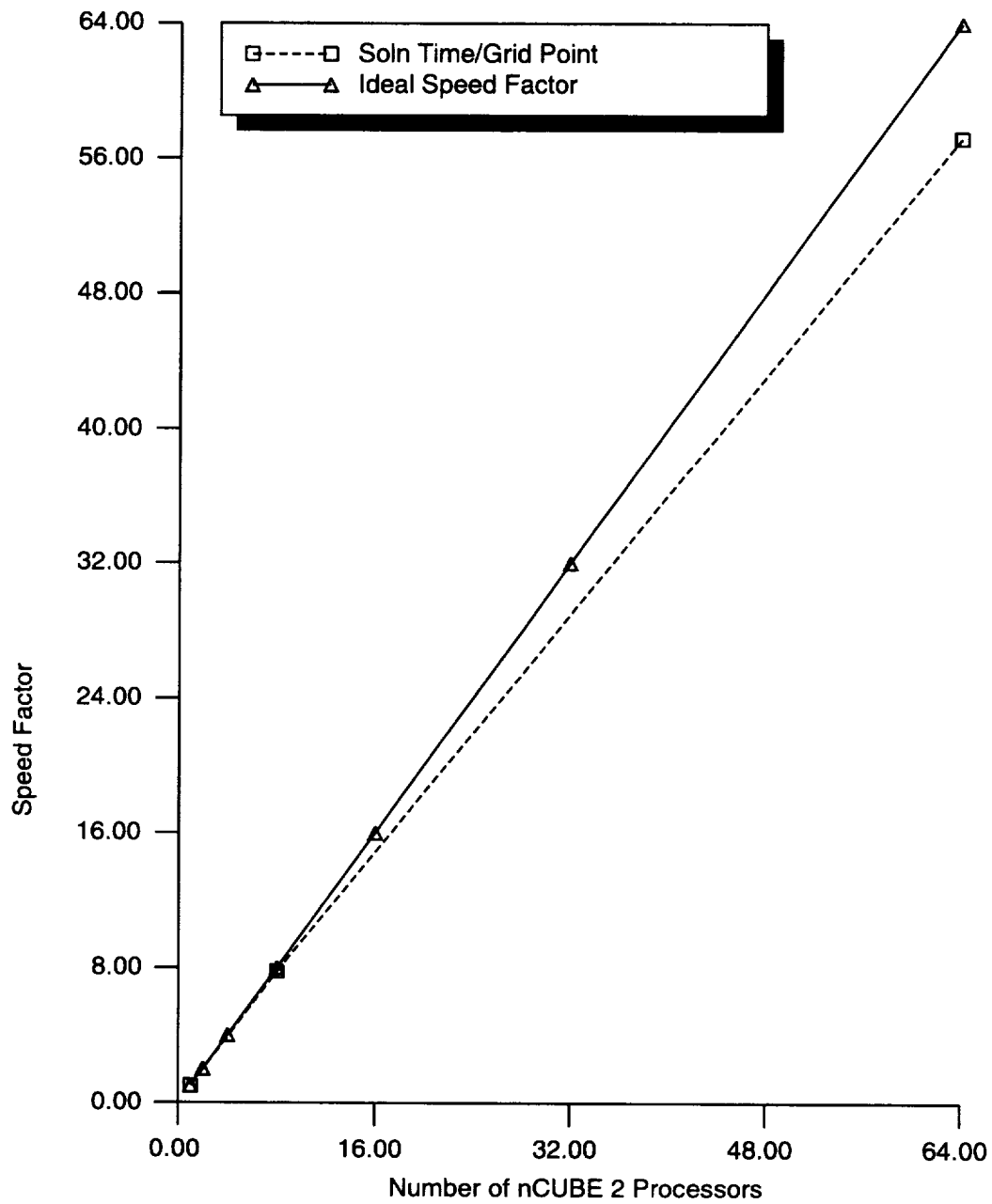


Figure 4.6: Comparison of speed factors for larger problems run on more processors.

***Parallel ADPAC Timing Runs
Successive Doubling of Problem Size***

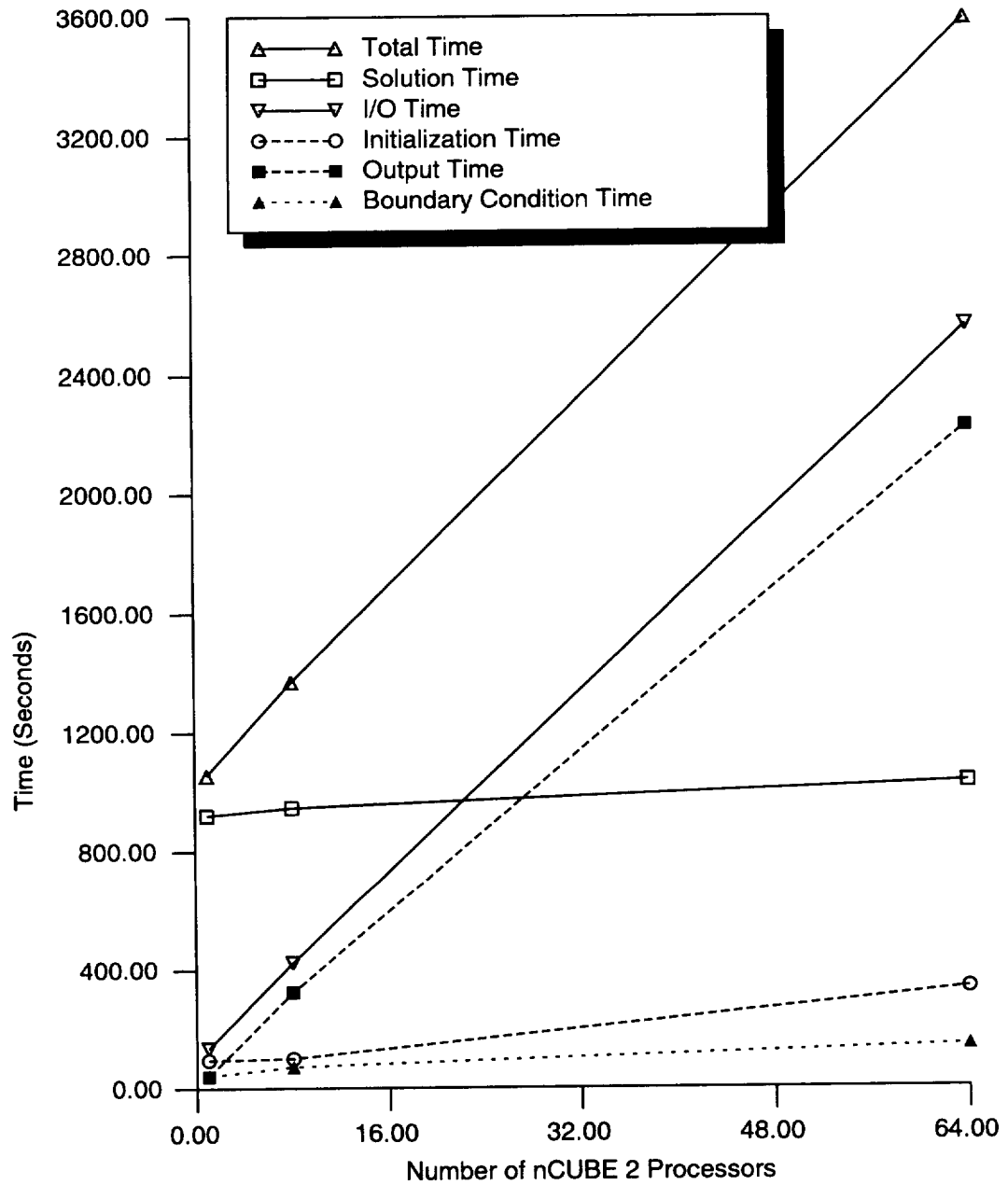


Figure 4.7: Comparison of CPU times for larger problems run on more processors.

The problem is not in the relative slowness of I/O due to problem size, but with the absolute slowness with which the output is performed. Using 64 processors, the restart and *PLOT3D* output uses 2/3 of the total CPU time for the run. There are some serial operations in the creation of the plot3d files which need to be parallelized better. Apparently, these serial operations are a severe impediment to good overall performance.

It has also been noted that I/O using the Scientific Database library (SDB) is not very efficient on many brands of machines. Previous experience has shown SDB to be a very poor performer on Cray computers, and it would appear that the same can be said for nCUBE machines. For example, a 3-D Euler solver using SDB input used 30% of the total CPU time to read the grid and write the restart file. The SDB performance was improved markedly when the FORTRAN conversion of each data word were scrapped in favor of direct C binary writes (see the discussion of CSDB in the *ADPAC07* Users Manual [28]).

LACE Cluster Results

Figure 4.8 shows the speed factors resulting from running the same problem on increasing numbers of IBM RS6000 workstations on the LACE cluster. In this figure, the same 64 block problem was run on a range of processors. The curve labeled "Total Time (LACE)" shows speed factors taken from wall clock times for each run. The second curve shows a data point using the ALLNODE switch. Although not evident on the graph, the IBM-proprietary ALLNODE switch (a special hardware device used for low latency interprocessor communication) provided a 30% performance improvement over using the standard ethernet network connection. An nCUBE curve from the successive halving is shown as a point of comparison.

Although this figure is interesting, it is not very significant. The LACE runs were made in a shared environment with contention for both CPU time and network bandwidth. The fact that the nCUBE curve is similar to the LACE curve is deceiving: the performance degradation is for different reasons. As explained above, the nCUBE speeds are degraded by input file growth and by a declining surface/volume ratio. The LACE runs were all on an identical problem: 64 blocks with the same boundary conditions in each run. The surface/volume ratio and the input file lengths are constant. The degradation on the LACE is a result of high latency in communications and contention for resources. Since contention was a significant problem (most calculations performed in multiple-user mode), no further analysis of these results was performed.

4.10 General Summary of Parallel Computing Experience Using *ADPAC07*

The following conclusions summarize the authors experience using the *ADPAC07* code in parallel computing environments. Parallel computations performed during this study utilized either a 64 processor nCUBE 2 parallel computer, or the NASA-Lewis LACE cluster computing environment.

Parallel/serial code flexibility can be adequately achieved using a block/processor decomposition approach.

Physically constructing a virtual parallel computer from networked workstations may require significant additional investment over and above the cost of the workstations alone (both in hardware and system administration).

The advantage of the networked workstation concept may lie more in the ability to

PRELIMINARY Parallel ADPAC Timing Runs
Constant Problem Size, Successive Doubling of Processors

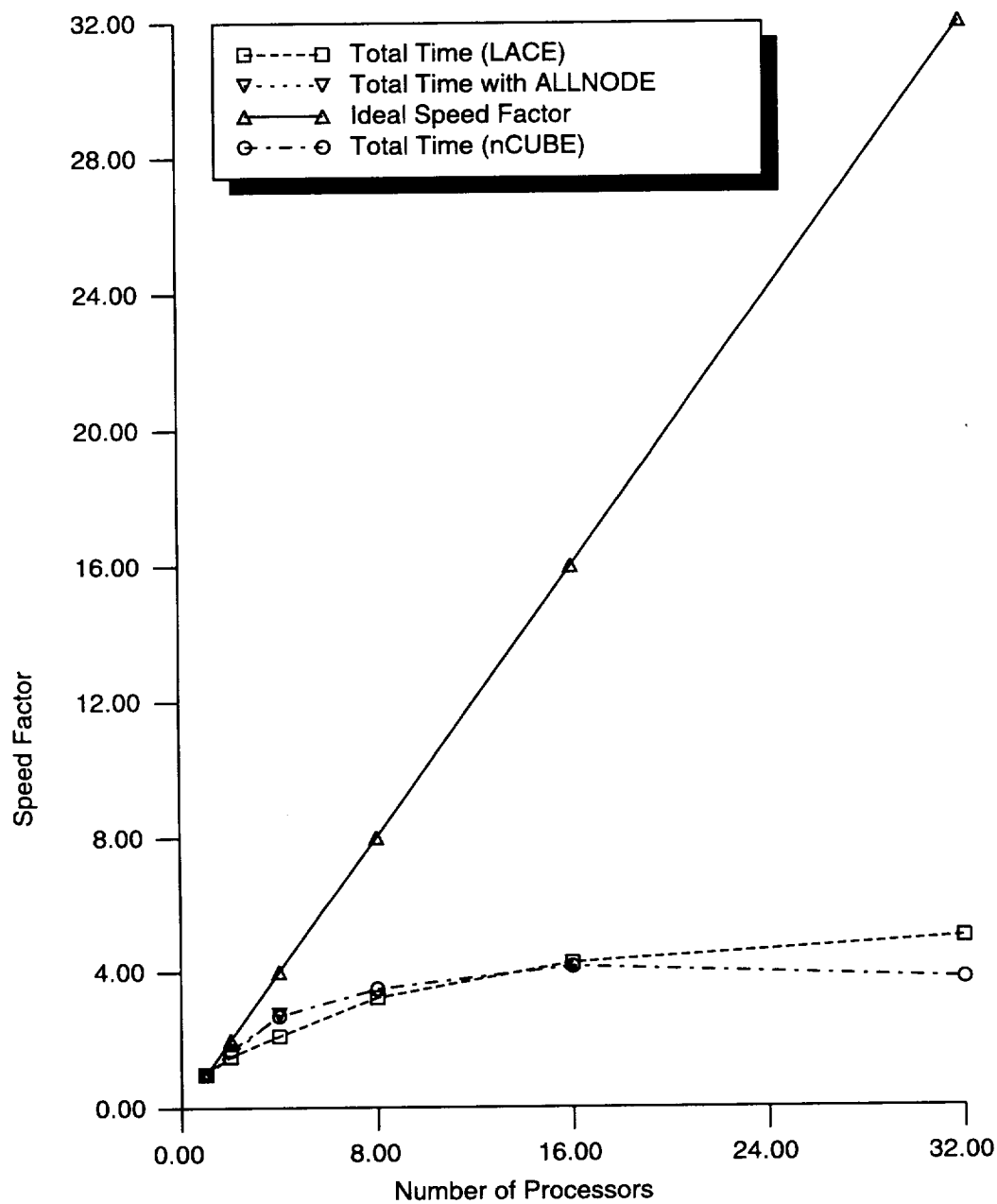


Figure 4.8: Comparison of speed factors for a single problem run on more RS6000 processors.

do extremely large problems in a reasonable amount of time as opposed to doing smaller problems very rapidly.

Like many studies of realistic applications involving parallel computing technologies, load balancing was identified as a critical issue for efficient use of networked computing resources.

Parallel computations are most effectively performed when the user carefully considers the effects of load balancing and minimization of interprocessor communication overhead through careful block/processor assignment and boundary condition specification. When these issues are carefully addressed, then there are significant advantages in employing the parallel capabilities of the *ADPAC07* code.

Chapter 5

ENDWALL TREATMENT ANALYSIS DESCRIPTION

In this section, the numerical techniques used to predict turbomachinery flowfields with endwall treatments are described. The general approach applied in this study was to exploit the multiple block mesh capabilities of the *ADPAC07* flow solver to couple the endwall treatment and blade passage flowfields. Separate computational domains (mesh blocks) were utilized for both the blade passage and at least one (or more) endwall treatment passages (grooves, slots, recessed vanes, etc.). Three specialized boundary conditions were developed to couple the flow domains. These boundary conditions result from various degrees of modeling assumptions used to simplify the blade passage/treatment passage aerodynamic interaction. Each of the three boundary conditions and the assumptions inherent to each approach are described in detail below.

5.1 Endwall Treatment Boundary Conditions

For the prediction of compressor endwall treatment flows, separate numerical mesh systems were utilized for both the compressor airfoil blade passages and the endwall treatment passages. Three different boundary condition procedures were utilized to couple the endwall treatment/blade passage flowfields and are illustrated graphically in Figure 5.1. The first technique, referred to as the direct-coupled approach, is utilized for those cases where there is a direct correspondence between mesh points in the treatment meshes and the blade passage meshes at the endwall interface. This construction permits a direct, point to point transfer of information from one mesh block to another. This approach is limited to geometries for which contiguous mesh systems can be generated, which, in this study, is limited to circumferential groove casing treatments.

The second approach, referred to as the endwall treatment time-average approach, is applied to endwall treatments which are non-axisymmetric in nature. This type of treatment includes discrete axial or blade angle slots, recessed vane sets, and circumferential grooves. The primary objective of this approach was to develop a numerical coupling scheme for discrete treatments which would represent the time-averaged influence of the treatment passage/blade passage aerodynamic interaction. This approach is restricted to steady state flows due to the time-averaged assumption inherent in the procedure, but simplifies the analysis in that only a single blade passage and a single discrete endwall treatment passage requires modeling in the overall solution. The endwall

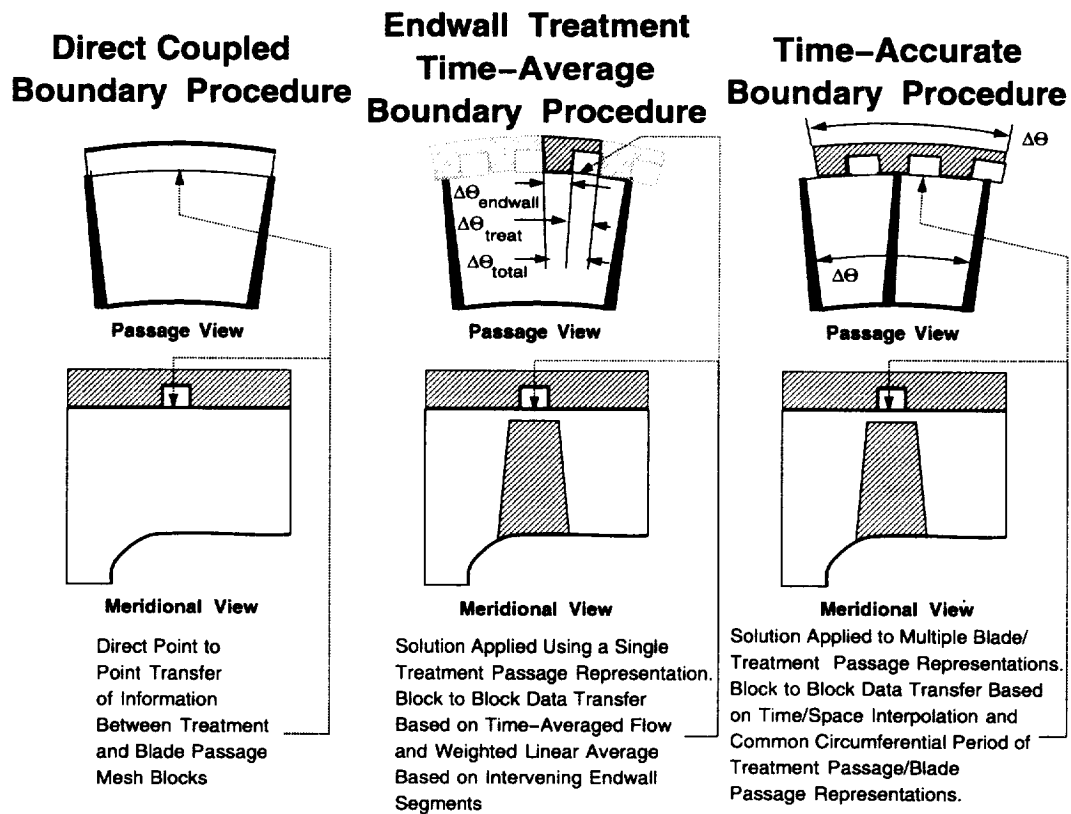


Figure 5.1: Endwall treatment numerical boundary conditions.

treatment time-average boundary treatment is based on a concept similar to the “mixing plane” treatment (see e.g. [35], [24]) often used in multistage turbomachinery flowfields. At the interface between the blade passage and the endwall treatment passage, flow data in each domain are circumferentially averaged (which represents a time average for a constant rotational speed, given the relative motion between the rotor and treatment passages), and passed to the neighboring domain as a boundary condition. The circumferentially averaged data from the treatment passage is modified to account for the finite intervals of endwall separating adjacent treatment passages from the rotor point of view. The boundary conditions for the blade passage are then constructed as a linear average of the circumferentially-averaged flow representation from the treatment grid and the known no-slip endwall boundary conditions. The linear average is based on the ratio of the circumferential extents of the treatment passage and intervening endwall. This scheme is illustrated graphically in Figure 5.1. Mathematically, for any variable ϕ , the boundary condition for the blade passage at the treatment interface is computed as:

$$\phi_{boundary} = \left(\frac{\Delta\theta_{treat}}{\Delta\theta_{total}} \right) \bar{\phi} + \left(\frac{\Delta\theta_{endwall}}{\Delta\theta_{total}} \right) \phi_{no-slip} \quad (5.1)$$

where $\phi_{boundary}$ is the boundary value applied for the blade passage, the $\Delta\theta$ terms are described on Figure 5.1, $\bar{\phi}$ represents a circumferential average of the data across the open area of the treatment, and $\phi_{no-slip}$ represents the appropriate no-slip boundary variables applied to the endwall portion of the overall treatment representation.

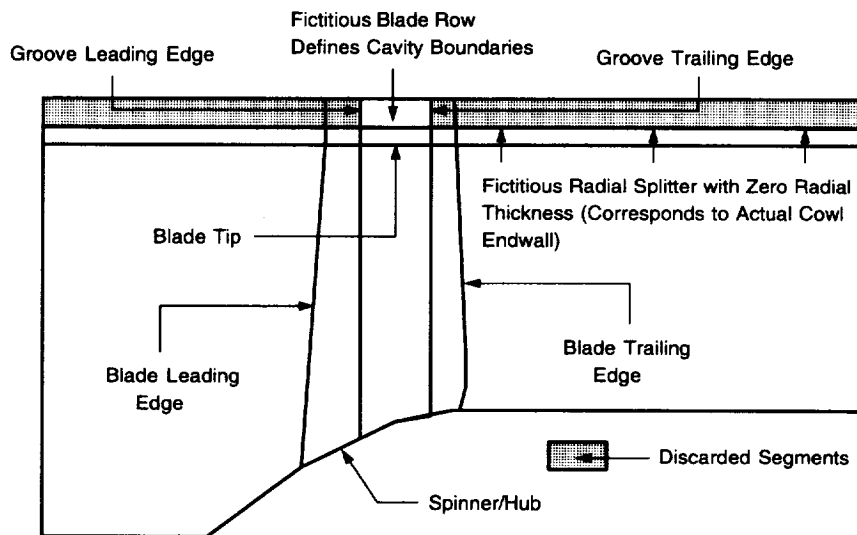
The third coupling procedure is referred to as the time-accurate coupling procedure. This procedure is utilized for detailed time-dependent solutions of the endwall treatment/rotor aerodynamic interactions, and is utilized in much the same way as rotor/stator interactions are computed in multistage turbomachinery flowfields. This approach was limited to treatment geometries which were reducible to small integer numbers of treatments per blade passage. This was done in order to minimize the circumferential extent of the rotor or treatment passages which must be modeled in order to employ a spatially periodic relationship for the overall blade passage/treatment representation.

5.2 Treatment Modeling Mesh Generation

All of the mesh systems utilized during this study were generated using the *TIGG3D* [36] mesh generation program. H-type meshes were utilized for both the blade passage and treatment flow regions. The H-type meshes simplified the transfer of information between the two flow fields as it was possible to generate the meshes with common axisymmetric interfaces across which circumferentially averaged or time-accurate circumferentially interpolated flow data could be transferred. A graphic illustration of the mesh generation procedure using the *TIGG3D* code is presented in Figure 5.2.

The treatment cavity was represented in the meridional plane as a fictitious blade row separated from the primary blade row by a fictitious radial splitter which corresponded to the actual endwall. This permits continuity of mesh lines across the actual endwall for both the primary and treatment blade passages. The portions of the mesh above the fictitious splitter which do not correspond to a treatment region were discarded. Once the meridional mesh was generated in this fashion, the circumferential mesh construction is performed by *TIGG3D*. Once a suitable grid was generated, the unused portions of the grid were discarded, and the final grid was assembled in a piecewise fashion. Treatment

Casing Treatment Mesh Generation Using *TIGG3D*



Most mesh systems used in this study were generated by *TIGG3D*. The treatment cavity is represented in the meridional plane as a fictitious blade row separated from the primary blade row by a fictitious radial splitter which corresponds to the actual endwall. The portions of the mesh above the fictitious splitter which do not correspond to a treatment region are discarded. Treatment mesh regions can also be created separately using an algebraic mesh generation program called *GROOVY*. The final mesh is then assembled in piecewise fashion.

Figure 5.2: Endwall treatment mesh generation using *TIGG3D*.

mesh regions were also created separately using an algebraic mesh generation program called *GROOVY* which utilized the meridional plane mesh point distribution generated by *TIGG3D*, but constructed a separate circumferential mesh point distribution for more complicated treatment cavities such as skewed axial slots.

5.3 *ADPAC* Solution Sequence for Endwall Treatment Flow-field Analysis

The solution for a fan rotor with casing treatment was normally initialized using the full multigrid start-up procedure with an exit static pressure which was lower than the design operating point. The purpose here was to choke the flow before attempting to resolve higher pressure ratios. Unless otherwise stated, all solutions were performed with 3 levels of multigrid, utilizing 50 iterations on each coarse mesh level before proceeding to the fine mesh.

When a converged solution was achieved for the initial back pressure, the solution was then recursively restarted by incrementally increasing the back pressure to obtain higher overall pressure ratios. The solution was often restarted not from the immediate previous solution, but one or two additional solution levels “upstream” of the present point on the operating curve (“upstream” implies previous solutions with lower exit static pressures). This was done to avoid artificially stalling the solution due to the sudden impulse of a step change in exit static pressure to achieve the new operating point. The procedure was repeated to obtain a constant speed operating characteristic for a given configuration. A figure illustrating this process is given in Figure 5.3.

At pressure ratios greater than the design pressure ratio, the solution was found to be very sensitive to back pressure, and the flow must be closely observed for signs of stall. Typically, stall was manifested as a divergent solution behavior, or through a constant, steady reduction in mass flow rate over a large number of iterations (1500 or so), which indicates that the phenomena (at least in this numerical context) is viscous, not pressure driven. Several difficulties were encountered in the prediction of the mass flow at stall. One problem was the dependence of the stall point on the numerical techniques used to obtain the solution (smoothing parameters, damping coefficients, multigrid, etc.). Another problem in the definition of the stall point is the long time scale required for the physical instability to evolve. Several solutions which seemed to be converged based on residual error, eventually stalled after a large number of iterations. These solutions exhibited a gradual migration in mass flow (while following the speed line) that eventually led to stall. Since each solution cannot be run for thousands of iterations due to economic considerations, a standard criteria needs to be developed to consistently define the stall point. In this study, solutions which did not display any signs of stall after 1000 iterations were deemed valid. Naturally, this criteria is admittedly rather vague and certainly dependent on the use of multigrid, time step size, etc. The authors believe that these problems are not unique to the *ADPAC* code and need to be addressed by the entire turbomachinery community.

Time-dependent solutions were generally initiated from a steady state solution with identical boundary conditions except for those which are inherently unsteady. Calculations were performed with both standard explicit and the implicit time-marching algorithms. Details for each calculation are given in the sections which follow.

Sample Iteration/Mass Flow Rate History for a Constant Speed Operating Characteristic Prediction

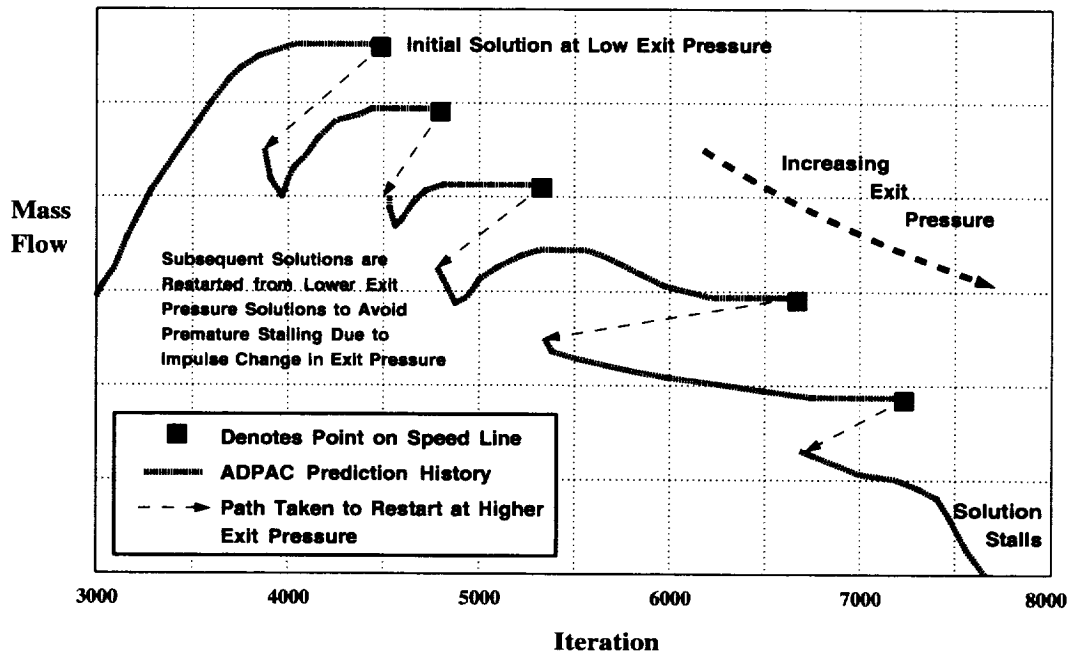


Figure 5.3: ADPAC sample iteration/mass flow rate history for a constant speed operating characteristic prediction.

Chapter 6

ADPAC07 VALIDATION TEST CASES

In this chapter, a number of validation test cases are described to illustrate and verify some of the computational enhancements to the *ADPAC07* code which were developed under this task order. These calculations more or less serve to lend confidence to aerodynamic predictions based on the *ADPAC07* code, in spite of the fact that many of the calculations are unrelated to the turbomachinery problems described by the objectives of this study. Specifically, algorithmical enhancements related to code parallelization, iterative implicit algorithm, non-reflecting boundary conditions, and two-equation turbulence modeling are validated by comparing predictions with existing data for a wide range of test cases. The test cases are grouped according to the feature being validated.

6.1 Turbulent Boundary Layer on a Flat Plate

The first case examined for verification of the $k - \mathcal{R}$ turbulence model was the prediction of a developing boundary layer along an infinite flat plate. The results presented here are representative of several flat plate boundary layer cases tested to examine the characteristics of the $k - \mathcal{R}$ turbulence model for a relatively well-known flow. The prediction of the flat plate flow was accomplished using algebraically generated H-type meshes with a fixed near wall spacing along the length of the plate. The freestream flow conditions and plate length were selected such that the plate length Reynolds number ($Re_x = \rho V x / \mu$) at the mesh outflow boundary was 3,000,000. Predictions were performed using both the algebraic (Baldwin-Lomax) turbulence model and the two-equation $k - \mathcal{R}$ model. The focus of this study was to determine how accurately the predicted turbulent boundary layer reproduced the well known log-law velocity profile behavior for zero pressure gradient flat plate boundary layer flows. In this respect, the velocity profiles are compared based on inner variables ($u^+ = u/v_{fric}$, $y^+ = yv_{fric}/\nu$, where $v_{fric} = \sqrt{\frac{\tau_{wall}}{\rho}}$). Figure 6.1 illustrates a comparison of predicted velocity profiles for both the algebraic and $k - \mathcal{R}$ turbulence models with the experimental data of Zukauskas [77] and the “law of the wall” turbulent velocity profile correlation (see e.g. White [79]). From this comparison, it is evident that both predictions yield reasonable results compared to both the experimental data and analytical correlations for both the inner (“law of the wall”) and the outer (“log law”) boundary layer regions. An analysis of variations with mesh density for these predictions is also included on Figure 6.1, and it is clear that the results were relatively mesh insensitive.

Zero Pressure Gradient Flat Plate Turbulent Velocity Profile

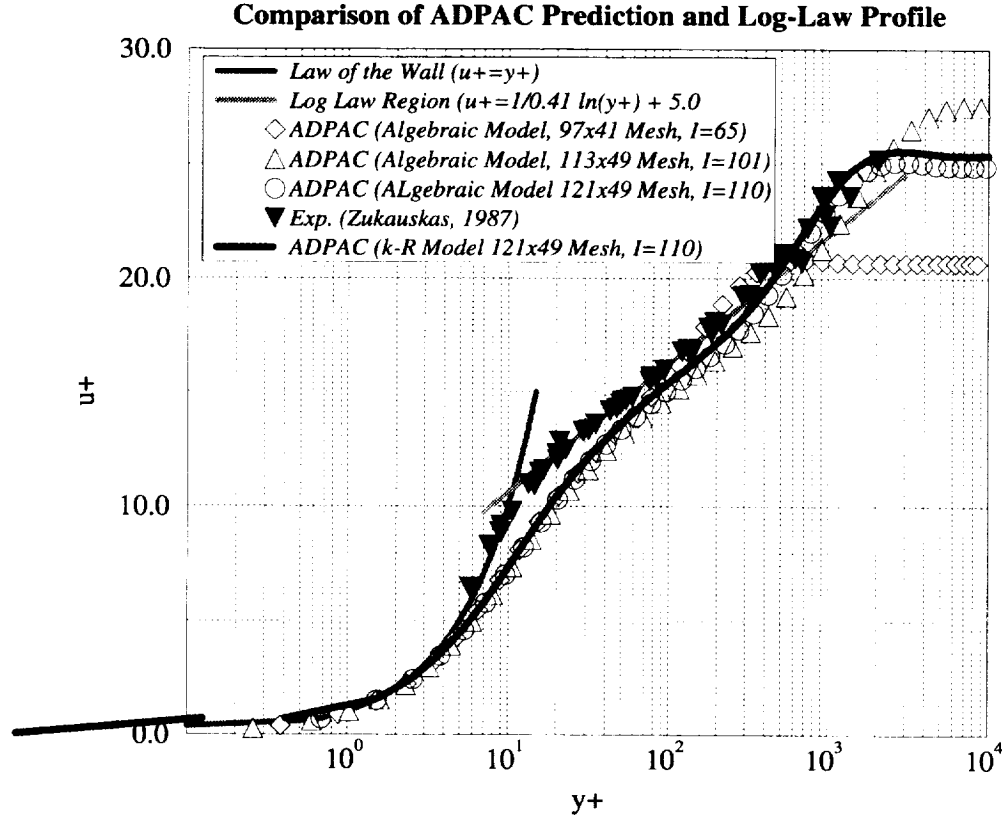


Figure 6.1: Flat plate turbulent boundary layer velocity profile comparison.

Also of interest in this case is the predicted wall surface skin friction coefficient ($C_f = \frac{\tau_{wall}}{\frac{1}{2} \rho u^2}$). Figure 6.2 displays both predicted, experimental, and turbulent boundary layer correlation for the skin friction coefficient distribution along the flat plate. Small discrepancies often occur in numerical predictions for flat plate skin friction coefficient distributions due to the inability to adequately simulate the analytical singularity which occurs at the flat plate leading edge. The end result of this anomaly is that the boundary layer often behaves as if the plate leading edge was shifted relative to the actual location. The predictions therefore appear slightly shifted on Figure 6.2. In spite of this phenomenon, generally good agreement was achieved for both the algebraic and $k - \mathcal{R}$ turbulence model predictions.

6.2 Bachalo and Johnson Transonic Bump Test Case

A geometrically simple, but aerodynamically complex transonic turbulent flow was reported by Bachalo and Johnson [71]. In this case, a high subsonic flow freestream ($M=0.875$) passes over a circular arc axisymmetric bump on an otherwise cylindrical centerbody. The geometry and idealized flow features are illustrated graphically on the inset diagram on Figure 6.4. The presence of the bump causes the flow to become transonic, and a relatively strong shock/boundary layer interaction occurs on the downwind portion of the bump. The shock is sufficiently strong to cause the boundary layer to separate, and subsequently

Turbulent Flat Plate Skin Friction Coefficient

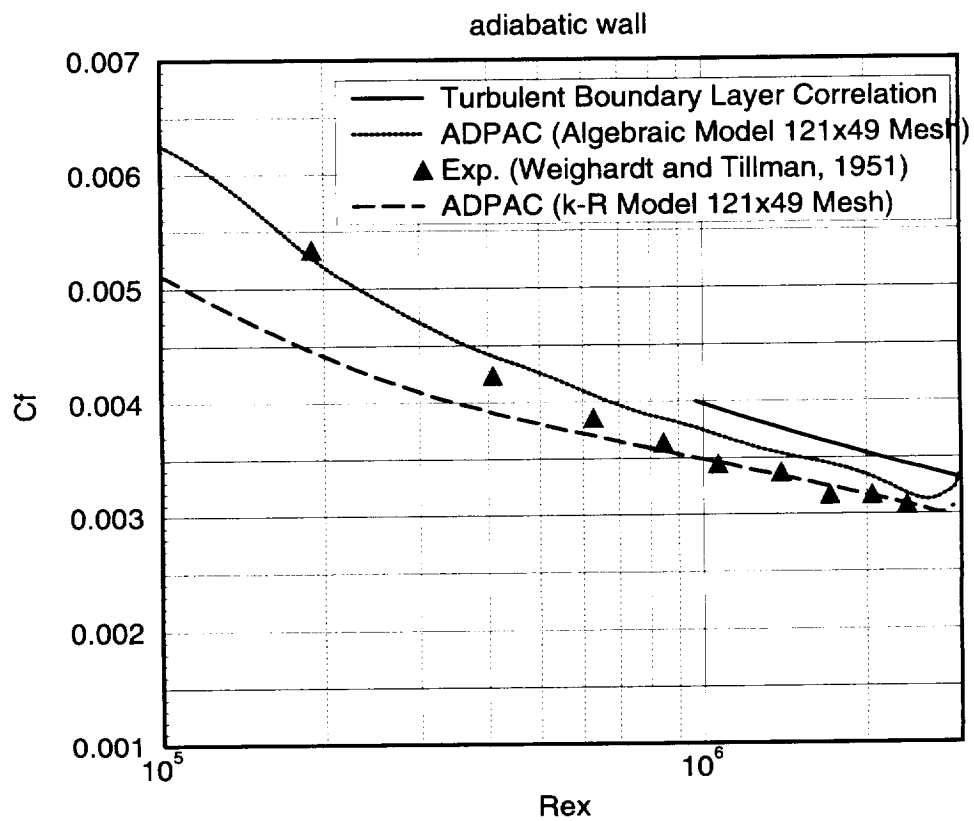


Figure 6.2: Flat plate turbulent boundary layer surface skin friction coefficient comparison.

reattach downstream of the bump. The separation region is particularly sensitive to the level of turbulence in the boundary layer, and the resulting pressure distribution along the bump and sting are highly dependent on the state of the boundary layer. This flow has proven to be a particularly challenging test case for numerical analysis as a result of the strong interactions which occur.

Prior to testing the $k - \mathcal{R}$ turbulence model for this flow, a series of calculations was performed using the axisymmetric flow solution capability of the *ADPAC* code on several different 2-D meshes using the algebraic (Baldwin-Lomax) turbulence model in an attempt to define the mesh sensitivity of this flow. The meshes were constructed algebraically using a straight H-type mesh topology. A fixed near wall spacing was employed along the bump and sting. A hyperbolic tangent stretching function was utilized to construct the remainder of the mesh between the primary geometric stations (inlet, exit, freestream, bump leading and trailing edges). The axial spacing along the bump was held constant.

Attempts to accurately predict this flow based on coarser meshes resulted in poor predictions either due to excessive smearing of the shock wave, or due to inadequate near wall spacing, or large mesh expansion ratios. The final mesh selected from the preliminary study is illustrated in Figure 6.3 and consisted of 161 axial and 73 normal mesh lines. The near wall physical mesh spacing was 0.001 cm, which resulted in an average near wall y^+ value of approximately 1.0. Figure 6.4 illustrates both the baseline (algebraic model) and two-equation ($k - \mathcal{R}$ model) predictions for the flow over the axisymmetric circular arc bump of Bachalo and Johnson using the 161x73 mesh. The figure illustrates both the predicted and experimental static pressure distributions along the downstream portion of the bump. The baseline *ADPAC* prediction using the algebraic model demonstrated reasonable agreement with the experimental data except for missing the slight dip in pressure near the recirculation region at $x/C=1.0$. The discrepancy in this area illustrates an area for improvement available through the use of advanced turbulence models. The *ADPAC* $k - \mathcal{R}$ turbulence model prediction for this flow more accurately reproduces the experimentally observed static pressure distribution, and one would subsequently assume that the prediction of the separation bubble was more accurate. Both the algebraic and $k - \mathcal{R}$ turbulence model bump surface static pressure predictions show a slight discrepancy in the position of the shock relative to the experimental data. This difference is believed to be due to uncertainty in the value of the upstream boundary layer thickness specified as input for the calculations.

An illustration of the predicted Mach number contours for both the algebraic and 2-equation turbulence model is given in Figure 6.5. Visually, the differences between the two solutions are slight, although a slight thickening and lengthening of the shear layer downstream of the bump is evident in the $k - \mathcal{R}$ model prediction when compared to the algebraic model prediction.

6.3 Mark II Turbine Vane Cascade

In order to assess the accuracy of the *ADPAC07* analysis of turbine vane heat transfer using the $k - \mathcal{R}$ turbulence model, predictions were performed for the turbulent flow over the Mark II planar vane cascade. The Mark II design is characteristic of an advanced first stage core turbine. The experimental data used for these preliminary comparisons were derived from Reference [78]. Experimental data were taken for two different exit Mach numbers (0.9 and 1.05) in a linear cascade facility. A complete description of the cascade

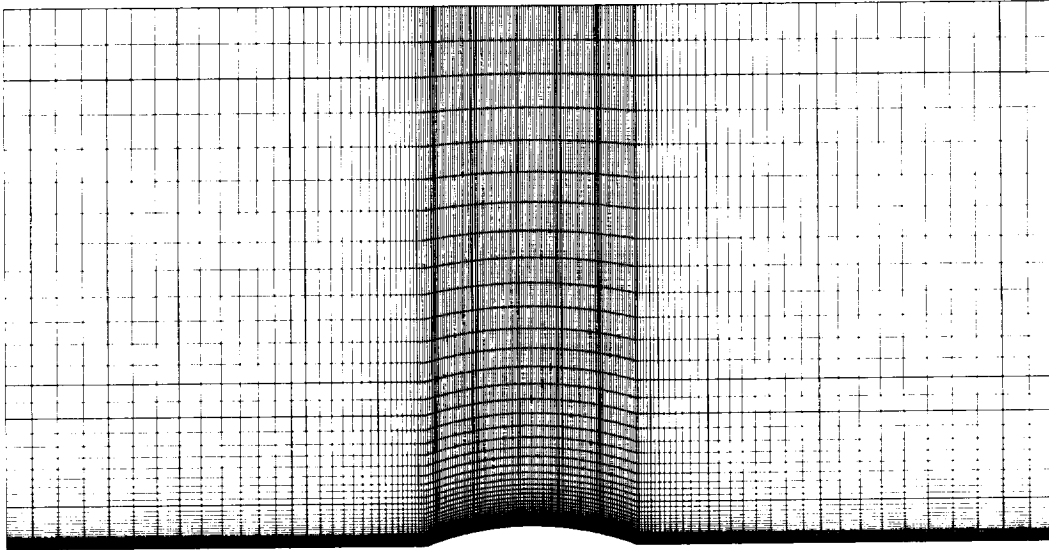


Figure 6.3: Bachalo and Johnson test case 161x73 2-D algebraic mesh.

Bachalo and Johnson Transonic Bump Test Case

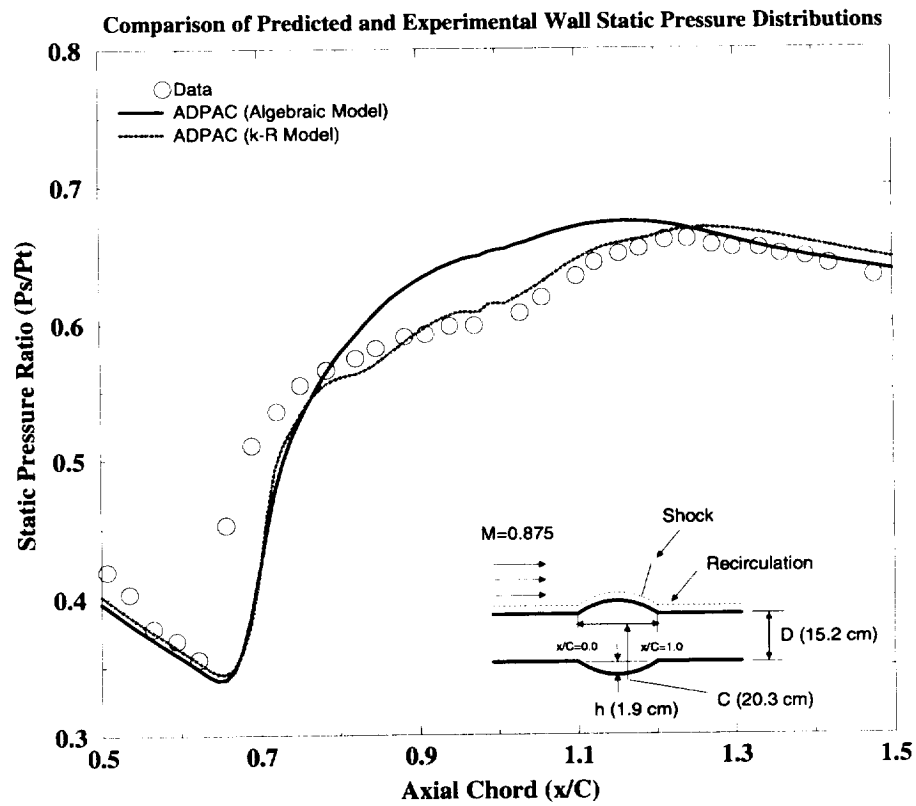


Figure 6.4: Comparison of predicted and experimental surface static pressure ratio for Bachalo and Johnson transonic axisymmetric bump test case ($M=0.875$).

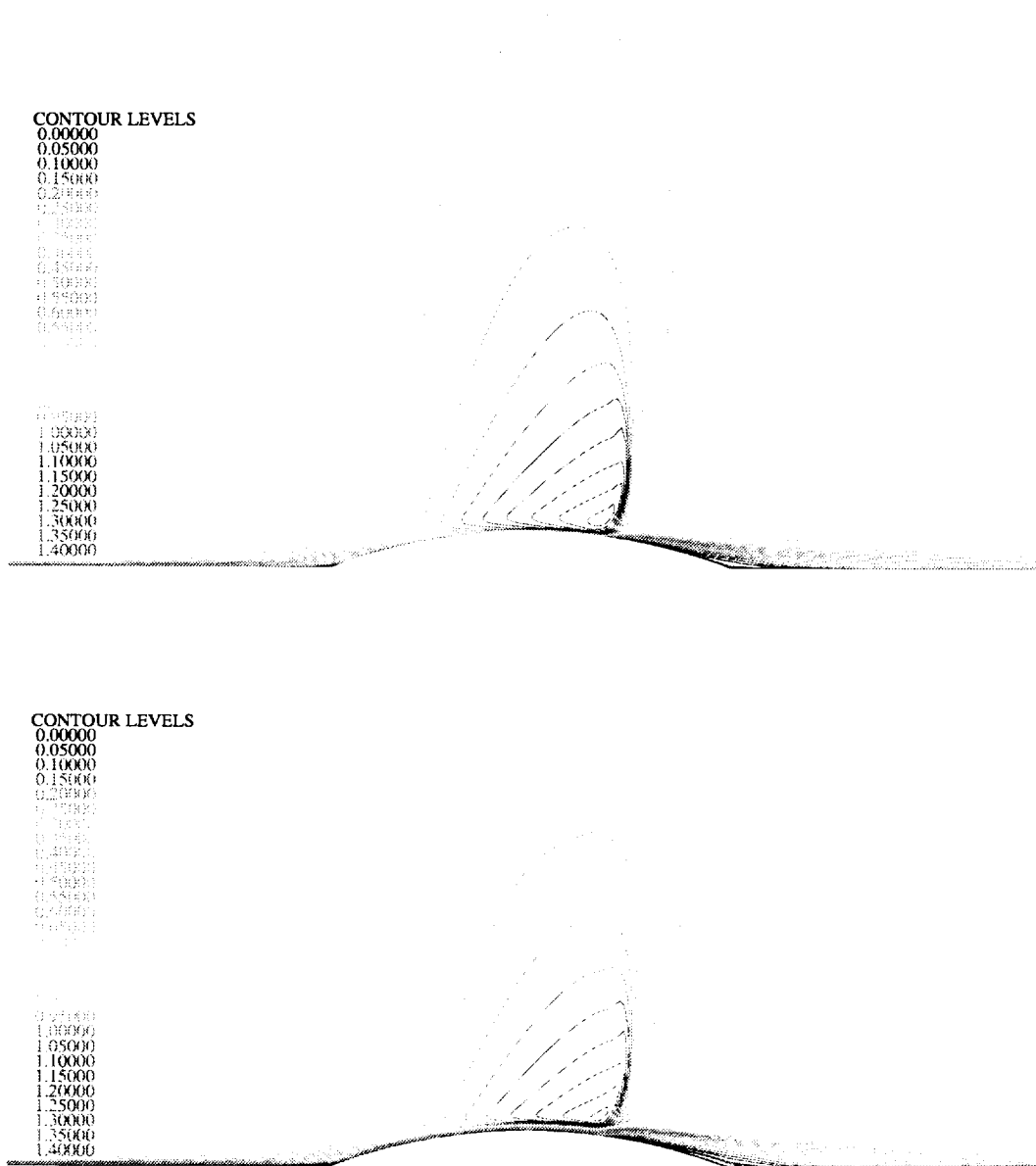


Figure 6.5: Comparison of predicted algebraic model (upper) and 2-equation model (lower) mach number contours for Bachalo and Johnson transonic axisymmetric bump test case ($M=0.875$).

Setting Angle	63.69 Degrees
Air Exit Angle	70.96 Degrees
Throat	1.568 inches
Vane Height	3.000 inches
Vane Spacing	5.108 inches
Suction Surface Arc	6.274 inches
Pressure Surface Arc	5.098 inches
True Chord	5.363 inches
Axial Chord	2.698 inches

Table 6.1: Mark II vane cascade design parameters.

facility and test procedure and data reduction are given in Reference [78]. Details of the Mark II vane design are given in Table 6.1.

An interesting feature of this test case is that at the exit flow Mach numbers tested, a strong normal shock forms on the suction surface of the airfoil at approximately 40% chord. This feature serves to illustrate the robustness of the $k - \mathcal{R}$ turbulence model in that a fairly strong shock-boundary layer interaction occurs and significant flow gradients are present, and yet, no special initialization procedures were required for the two-equation turbulence model solution.

The *ADPAC07* analysis was applied to the Mark II vane to predict both aerodynamic and heat transfer performance. Since this geometry was analyzed extensively under a previous study [25], several mesh systems and predictions using the algebraic turbulence model were already available for comparison. A 193x33 2-D C-type mesh was used for the $k - \mathcal{R}$ model calculation, and is illustrated graphically in Figure 6.6. This mesh employs a noncontiguous mesh along the C-grid cut boundary to avoid the problems of excessive grid shear often associated with contiguous C-type meshes for turbomachinery blade rows. Additional details related to the mesh generation, mesh dependence study, and algebraic model aerodynamic and heat transfer predictions for the Mark II vane are available in Reference [78].

The flow conditions selected for this test case correspond to Run 4321 of the Mark II airfoil described in [78]. The exit Mach number was 0.89, inlet total pressure and total temperature were 38.33 psia and 1389 degrees Rankine, respectively. A comparison of the experimental predicted airfoil surface static pressure distributions is given in Figure 6.7. Both the algebraic and two-equation model predictions are plotted. The previously mentioned shock wave is quite apparent in the large pressure gradient on the suction surface of the blade. Both numerical predictions displayed outstanding agreement with the experimentally-derived airfoil surface static pressure distributions. A comparison of the experimental predicted airfoil surface heat transfer coefficient distributions is given in Figure 6.8. Again, both the algebraic and two-equation model predictions are plotted. In

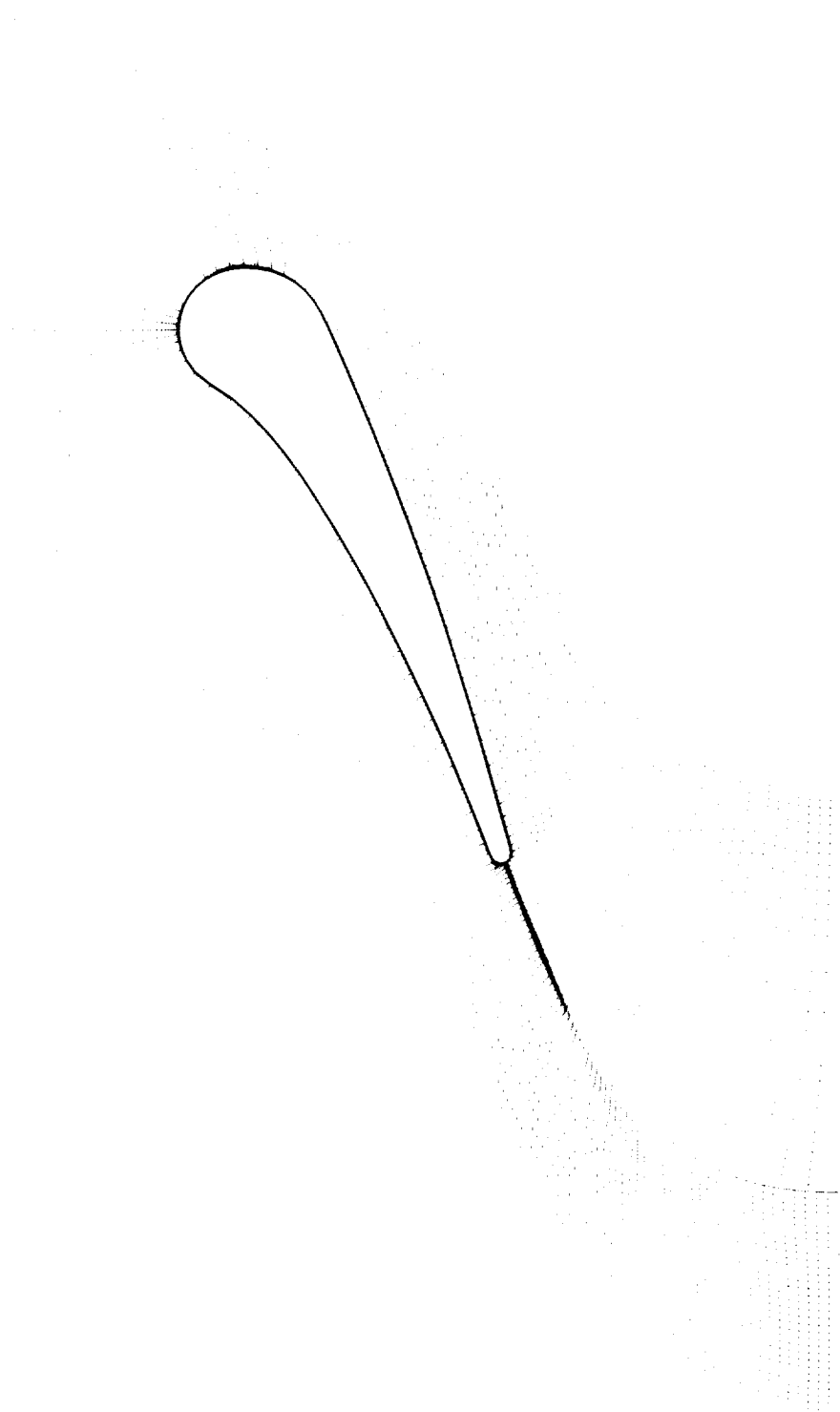


Figure 6.6: 193x33 2-D mesh system for Mark II turbine vane cascade.

this case, significant differences in the predictions exist due to the absence of a transition model in the $k - \mathcal{R}$ results. The algebraic model results employed a “point” transition model (instantaneous transition at a point defined by the model based on the maximum predicted turbulent viscosity across the boundary layer) specifically developed for turbine airfoil calculations, and hence, provides better agreement with the experimental data along the suction surface. On the other hand, proper specification of the freestream values for the k and \mathcal{R} model variables permits the simulation of variations in freestream turbulence intensity ($Tu=6.5\%$ in this case). This capability has the effect of improving the predictions along the pressure surface of the vane. The two-equation model overpredicts heat transfer in the near stagnation region and these results closely match the “fully” turbulent results previously obtained with the algebraic model without transition (see [25]). In any case, the two-equation model results without transition model are not unreasonable, and further application of this model with improved transition characteristics would likely yield even better results.

6.4 VBI Vane Test of Non-Reflecting Boundary Conditions

The purpose of this test cases was to determine the effectiveness of the quasi-3D non-reflecting boundary conditions described in Chapter 3. The vane geometry from the turbine Vane-Blade Interaction (VBI) [80] rig was selected as a test case for the non-reflecting boundary conditions. The exit Mach number for the case tested was 1.12. This geometry and flow condition provided a formidable test for the non-reflecting boundary conditions, particularly when the exit boundary plane is close to the vane due to the formation and propagation of a shock off the vane trailing edge.

A series of steady-state Euler calculations were initially performed on the geometry using an O-type mesh around the vane generated with an elliptic mesh generation scheme. The mesh consisted of 46,529 points ($161 \times 17 \times 17$). This relatively coarse grid was selected to permit rapid job turnaround, and because previous calculations had shown that this mesh was sufficient to capture the experimentally observed data trends. The boundaries of the vane O-grid, referred to as the *short* grid, are outlined in the center of Figure 6.9. H-grid extensions were added upstream and downstream of the O-grid to move the boundary conditions approximately one chord length away from the vane. This composite grid, also shown in Figure 6.9, is referred to as the *long* grid.

Four numerical configurations were tested: both short and long grid using the previous turbomachinery (Reimann invariant) boundary conditions, and both short and long grid using the non-reflecting boundary conditions. Each configuration was run using identical inlet profiles and exit static pressure at the hub to give the proper design exit mach number at the midspan of the vane. The data presented from the converged solutions includes vane exit data plane Mach number distributions and midspan vane surface static pressure distributions. Figure 6.10 illustrates the vane exit data plane Mach number contours from the four test configurations. The location of the vane exit data plane was identical for all four cases and is shown in Figure 6.9. The solutions employing the long grid show very little difference between the original turbomachinery boundary condition results (Figure 6.10a) and the non-reflecting boundary condition results (Figure 6.10b). This indicates that the inlet and exit boundaries were sufficiently removed from the vane such that the boundary condition type did not affect the solution.

However, when the grid extensions are not used (short grid) and the boundary condi-

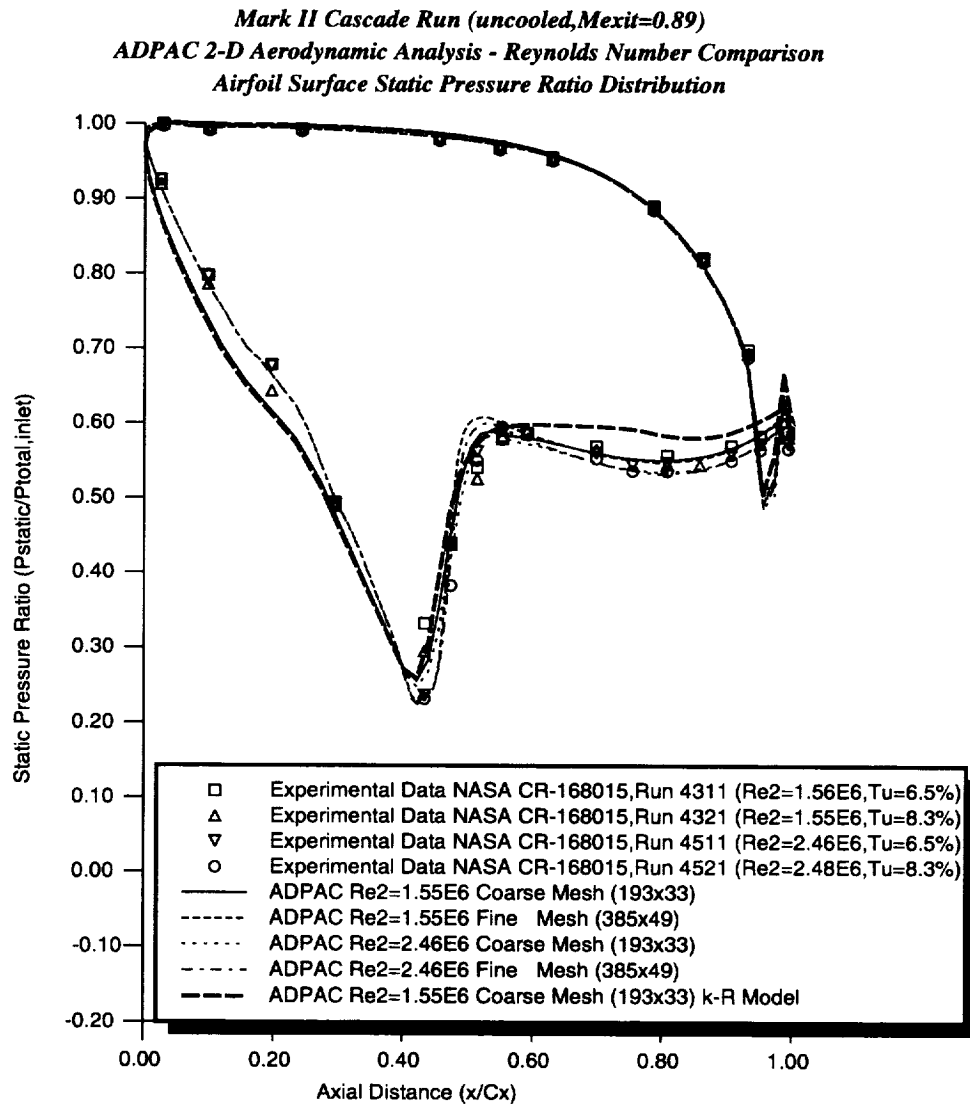


Figure 6.7: Comparison of experimental and predicted airfoil surface static to inlet total pressure ratio for Mark II turbine vane cascade.

Mark II Airfoil Cascade (uncooled, Mexit=0.89)
ADPAC 2-D Aerodynamic Analysis - Reynolds Number Comparison
Airfoil Heat Transfer Coefficient Distribution ($h_o=200 \text{ BTU/hr/sqft/F}$)

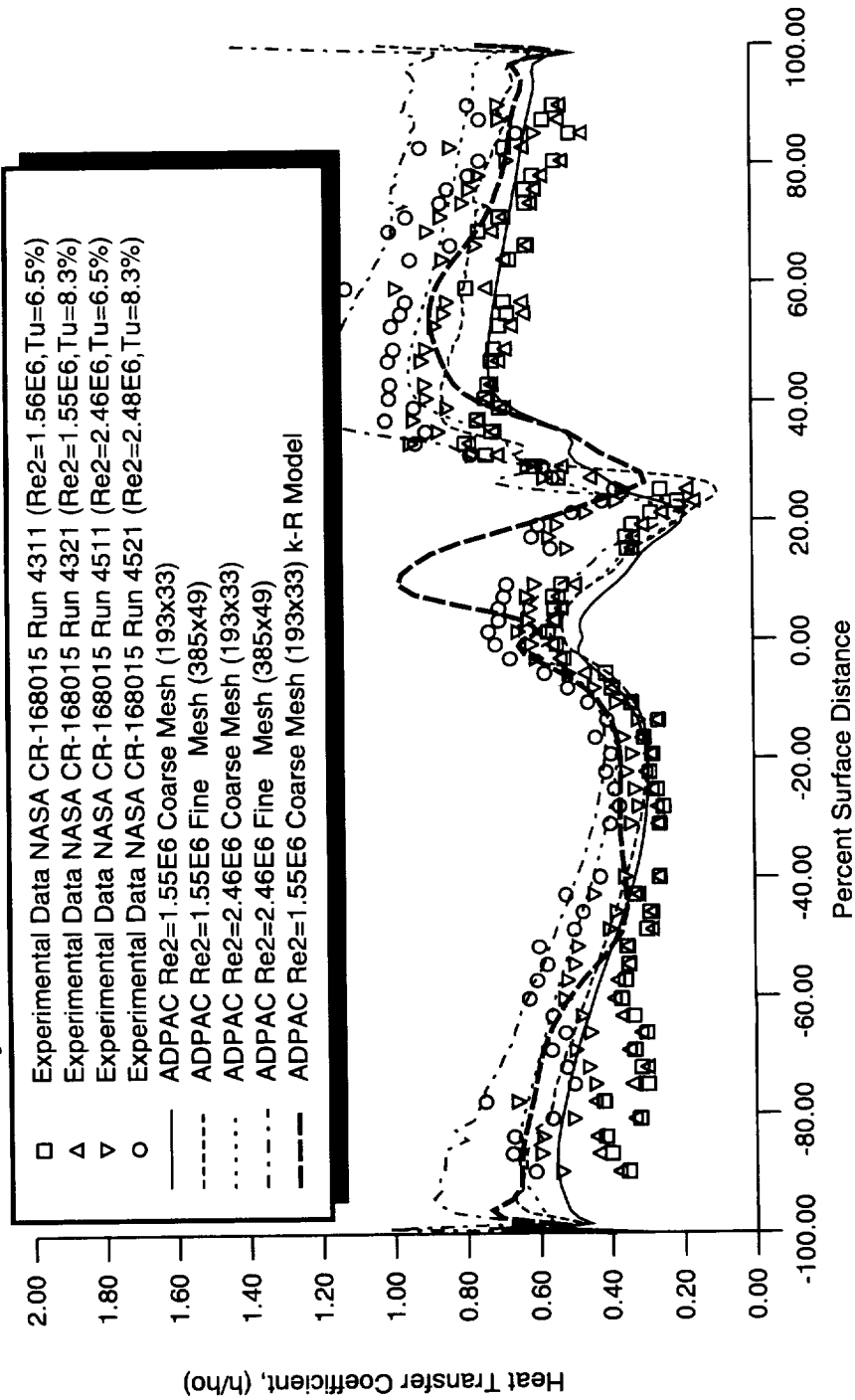


Figure 6.8: Comparison of experimental and predicted airfoil surface heat transfer coefficient for Mark II turbine vane cascade.

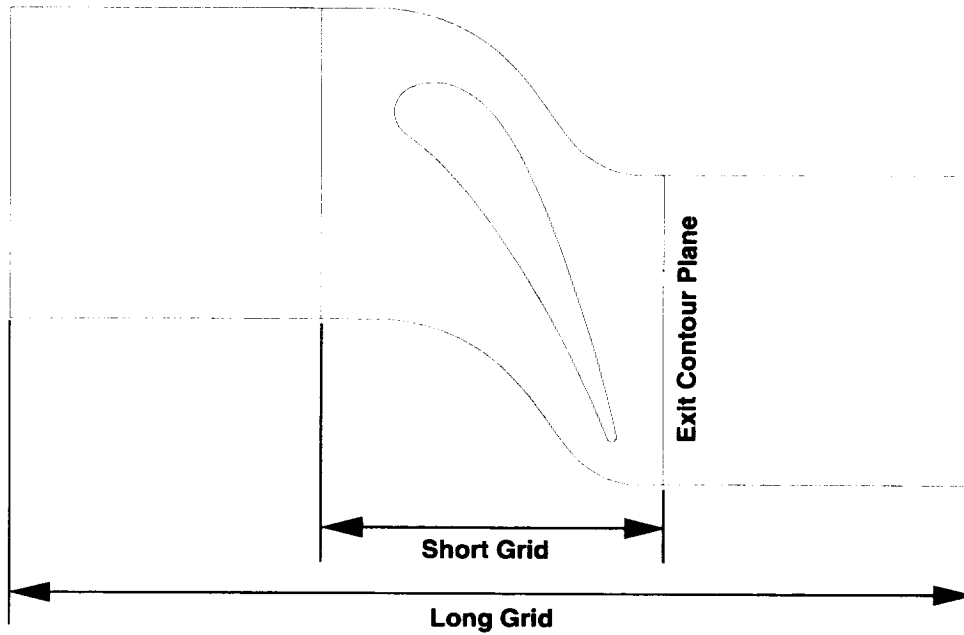


Figure 6.9: VBI vane short grid and long grid boundary outlines and vane exit data plane location.

tions are applied at the inlet and exit of the O-grid, the difference between the two boundary conditions becomes much larger. The original turbomachinery (Reimann invariant) boundary conditions partially constrain the exit static pressure based on the hub static pressure specification and application of radial equilibrium along radial mesh lines. This effectively smears out any large gradients in pressure, such as a shock, at the bounding plane (Figure 6.10c). The new non-reflecting boundary conditions allow variations in pressure across the pitch of the blade while constraining the *average* pressure at each radial slice to satisfy the radial equilibrium equation. When applied on the short grid, the non-reflecting boundary conditions yield Mach number distributions at the vane exit data plane (Figure 6.10d) which show great similarity with the long grid solutions, and clearly show the resolution of the vane trailing edge shock passing cleanly through the exit boundary plane.

The location and type of boundary condition used also affected the predicted vane surface static pressure distribution. The effect of the exit boundary scheme was most evident on the suction surface of the vane near the trailing edge. Non-dimensional distributions of vane surface static pressure taken at the midspan of the vane from the four test solutions are compared with experimental data in Figure 6.11. The data are presented as a function of wetted distance along the vane surface from the geometric leading edge. The two long mesh solutions show good agreement with each other and reasonable agreement with the experimental data. Predicted results from the short grid using the original boundary conditions show large variations in the pressure distribution near the suction surface trailing edge. The oscillatory behavior of this solution shows poor qualitative agreement with the experimental data and the long grid solutions, presumably due to the imposed exit plane static pressure and the resultant effect on airfoil loading. Application of the non-reflecting boundary conditions for the short grid solution resulted in a vane surface pressure distribution which matches much more closely the long grid solutions both qualitatively and quantitatively.

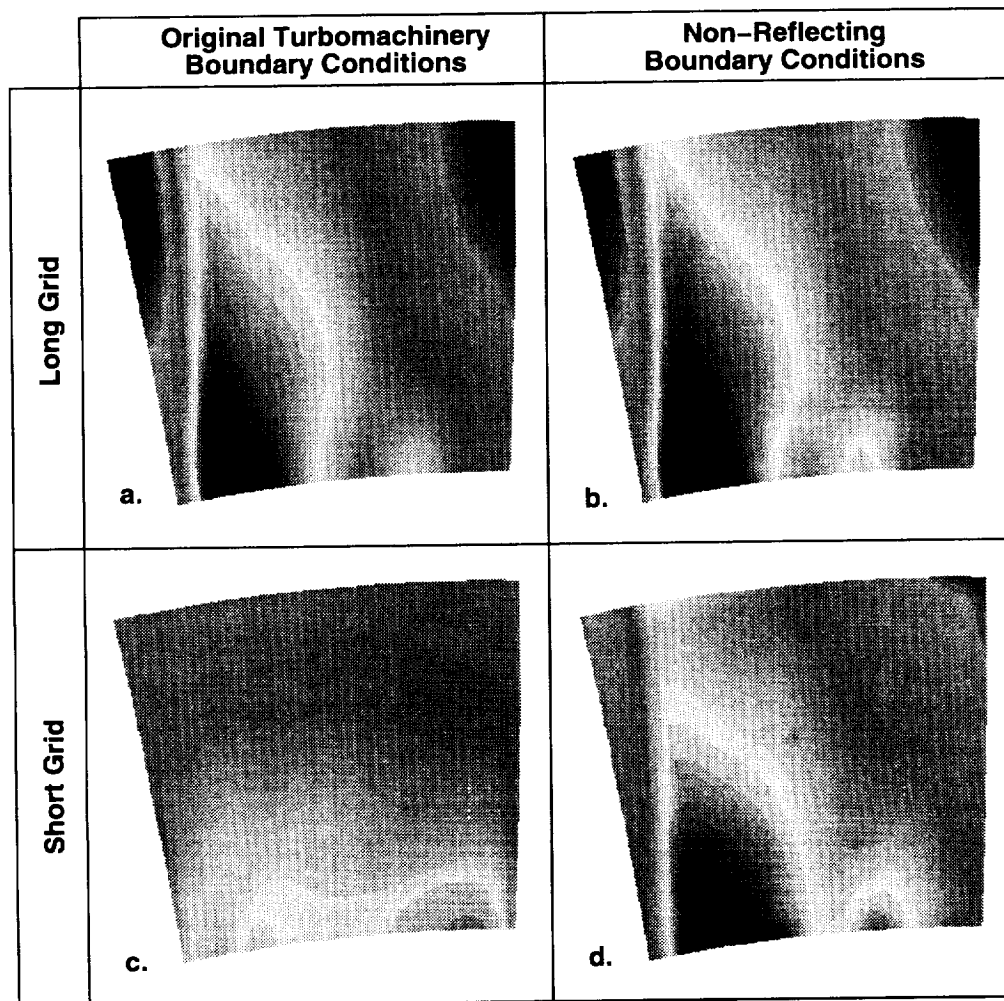


Figure 6.10: Exit plane Mach number contours for the VBI vane for various grid lengths and boundary condition combinations.

VBI Vane Midspan Surface Pressure Distribution

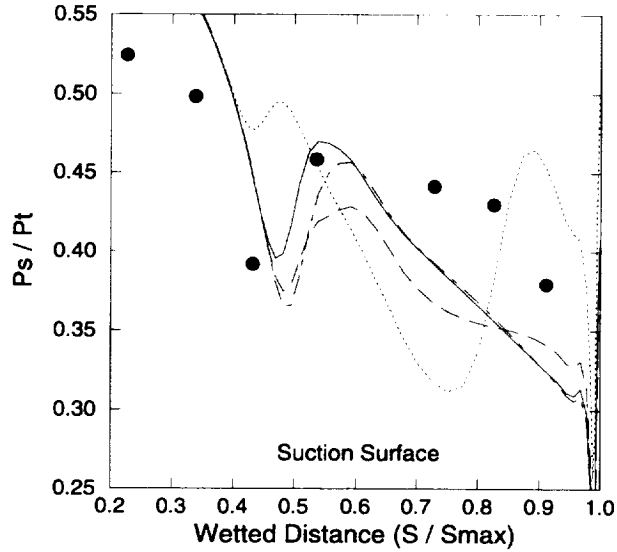
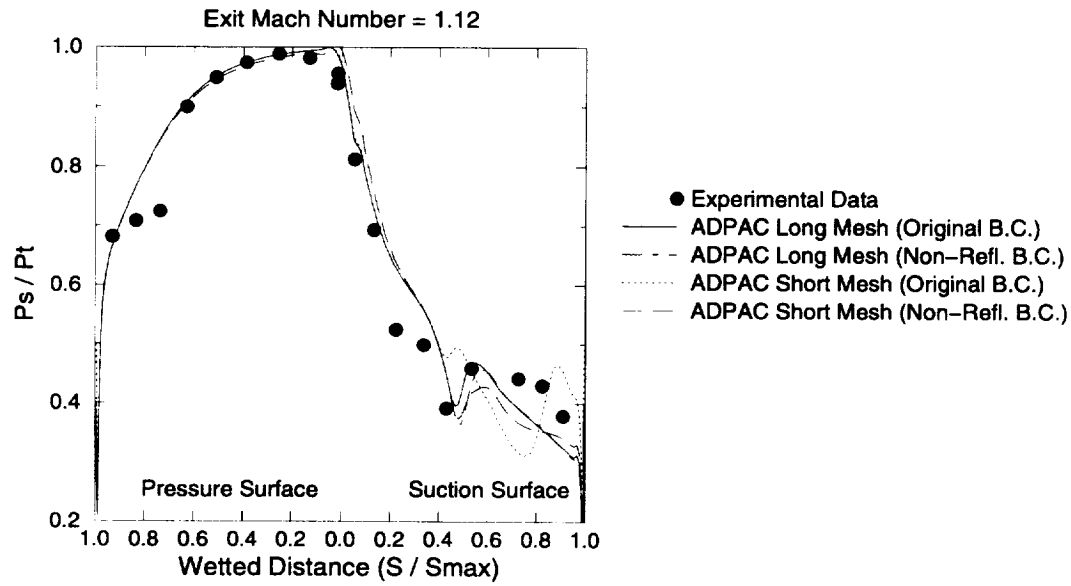


Figure 6.11: Inviscid vane surface static pressure distributions along the midspan of the VBI vane.

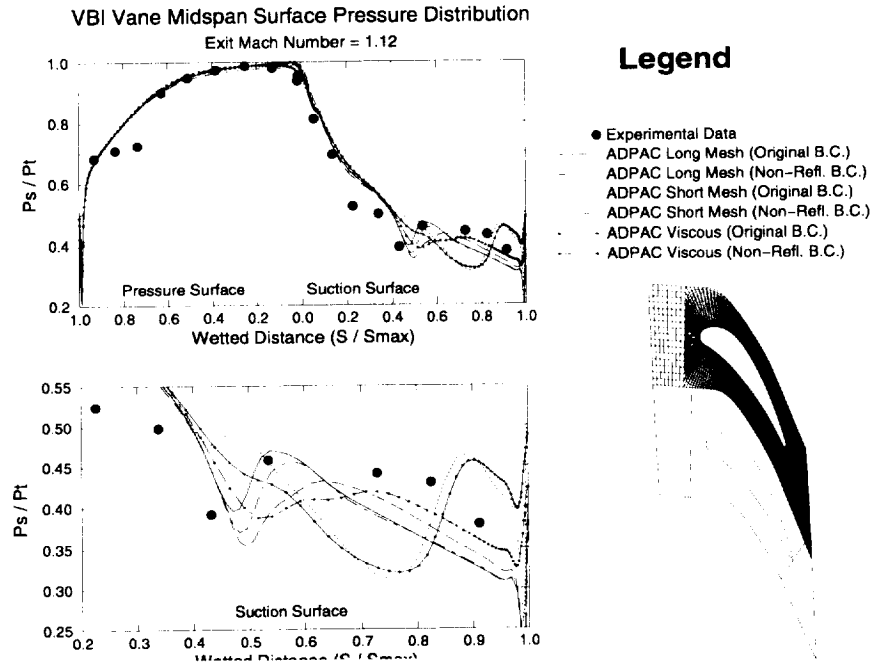


Figure 6.12: Comparison of inviscid and viscous predicted midspan vane surface static pressure distributions for VBI vane.

The results presented above for the VBI vane were based on a series of inviscid (Euler equation) calculations comparing the use of the standard, Reimann invariant *ADPAC* turbomachinery boundary conditions, and the recently developed non-reflecting boundary conditions. The results described below describe a similar comparison based on viscous (Navier-Stokes equations) solutions.

Steady state Navier-Stokes calculations were performed for the VBI vane using a non-periodic O-type mesh around the vane generated with an elliptic mesh generation scheme. H-type mesh extensions are provided upstream and downstream of the blade as shown in Figure 6.12. The overall mesh consisted of roughly 265,500 points.

An illustration of the predicted (both Euler and Navier-Stokes) and experimental (time-mean of time-dependent measurements) vane midspan surface static pressure ratio distributions are given in Figure 6.12. Calculations using both the standard turbomachinery boundary conditions and the non-reflecting boundary conditions are presented. In spite of the difference in mesh systems, and mesh extent, both the Euler and Navier-Stokes predictions using the standard turbomachinery boundary conditions were in good agreement with each other. Unfortunately, the calculations with the standard turbomachinery boundary conditions do not correlate well with the experimental data in the vicinity of the vane suction surface near the trailing edge (see expanded scale region on Figure 6.12). The reason for this discrepancy is apparent in the midspan blade-to-blade and exit plane Mach contours and pressure contours given in Figures 6.13 and 6.14, respectively. In order to satisfy the constant static pressure and radial equilibrium conditions imposed by the standard turbomachinery boundary conditions, a compression process is set up on the vane suction surface

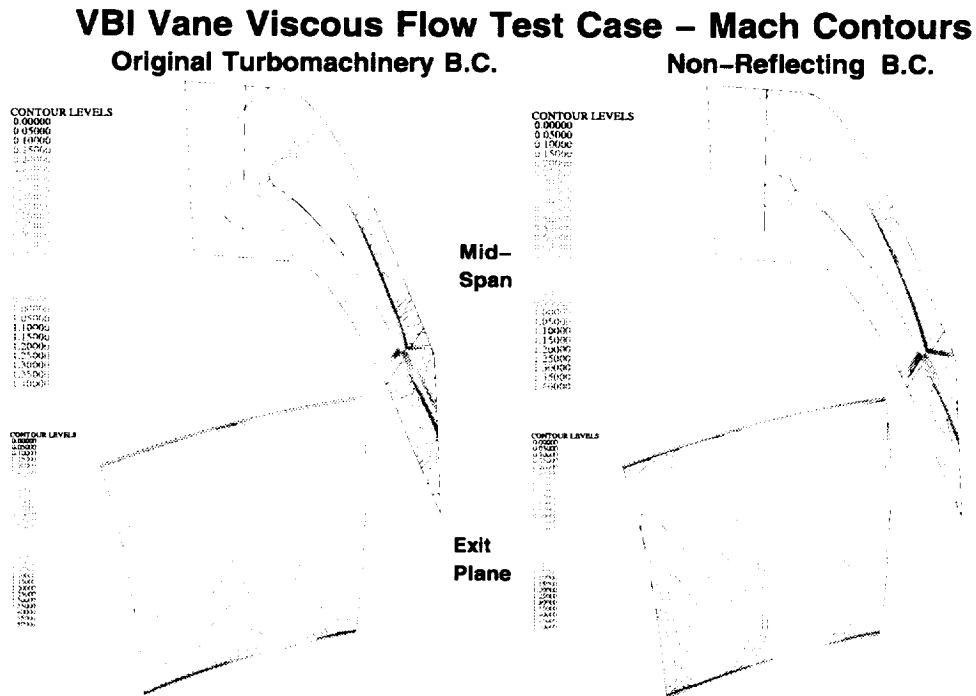


Figure 6.13: Comparison of midspan blade to blade plane and exit plane (1/4 chord aft of trailing edge) predicted Mach contours for the VBI vane.

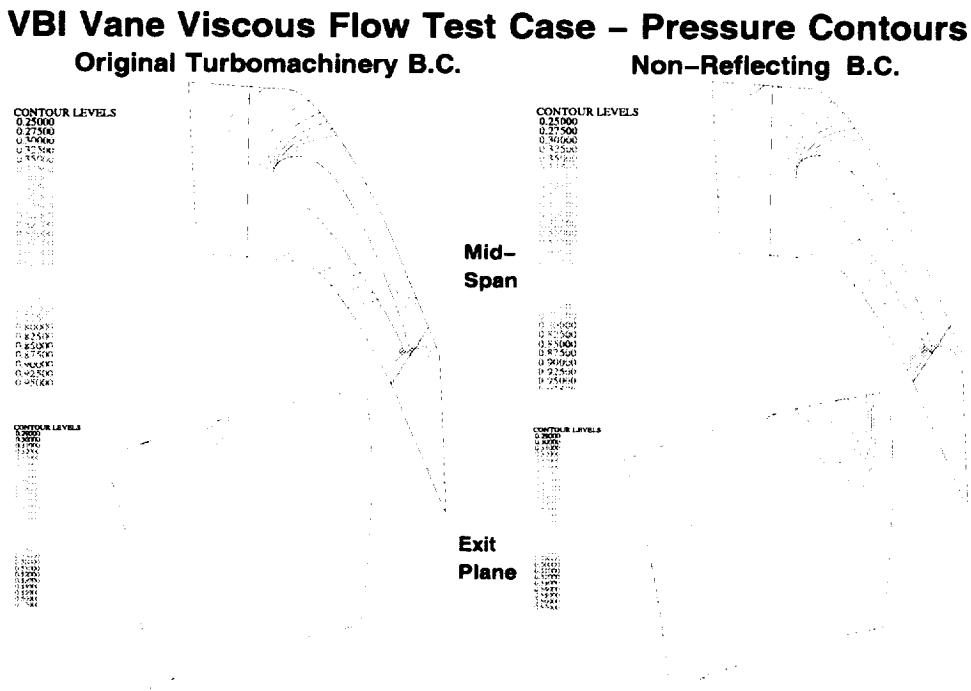


Figure 6.14: Comparison of midspan blade to blade plane and exit plane (1/4 chord aft of trailing edge) predicted static pressure ratio contours for the VBI vane.

ADPAC Implicit Algorithm Test Case:
Vortex Shedding Behind a Circular Cylinder in Crossflow

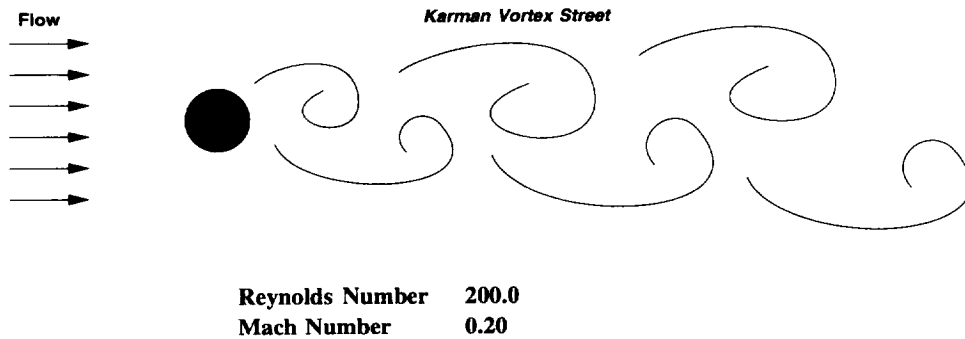


Figure 6.15: Schematic illustration of vortex shedding behind a circular cylinder in crossflow test case (Reynolds number = 200.0).

which causes the rapid change in static pressure near the nondimensional wetted distance of 0.9 (see Figure 6.12) for the standard boundary condition calculations. In every mesh and flow solution tested, the non-reflecting boundary conditions show improved agreement with the experimental data (see Figure 6.12), and permit unaltered propagation of the trailing edge shock pattern through the downstream boundary (see Figures 6.13 and 6.14). It should be noted that the overall solution convergence and predicted mass flow and total pressure loss for the viscous solutions were essentially unchanged by the choice of boundary condition.

These results validate the implementation and highlight the improvements gained from the use of non-reflecting boundary conditions. The number of points in the short grid resulted in a nearly 25% reduction when compared to the number of points in the long grid, while the quality of the results remained essentially constant.

6.5 Vortex Shedding Behind a Circular Cylinder in Crossflow

The results presented in this section are based on the prediction of the two-dimensional flow over a circular cylinder in crossflow. This test case is illustrated schematically in Figure 6.15. The flow conditions were chosen (Reynolds number = 200 (based on cylinder diameter)) such that vortex shedding was expected to occur downstream of the cylinder. This simulation provides a geometrically simple, yet aerodynamically complex test for the *ADPAC07* implicit time-accurate flow solver. The flow Mach number was set at 0.2, and the solution was initialized from a converged steady state solution at a Reynolds number of 50.0 (for which vortex shedding is not expected to occur) and a 3 degree angle of attack. To start the shedding run the boundary conditions are set to obtain zero angle of attach inflow and a Reynolds number of 200.0 (the asymmetry of the initial flow, or sudden angle of attach change, leads to the vortex shedding behavior). Solutions were obtained for various fixed time increments resulting in approximately 40, 80, and 160 implicit time steps per oscillation cycle using both the original and modified iterative implicit algorithms described in the previous chapters. A 2-D polar grid employing 161 normals and 65 contours (see

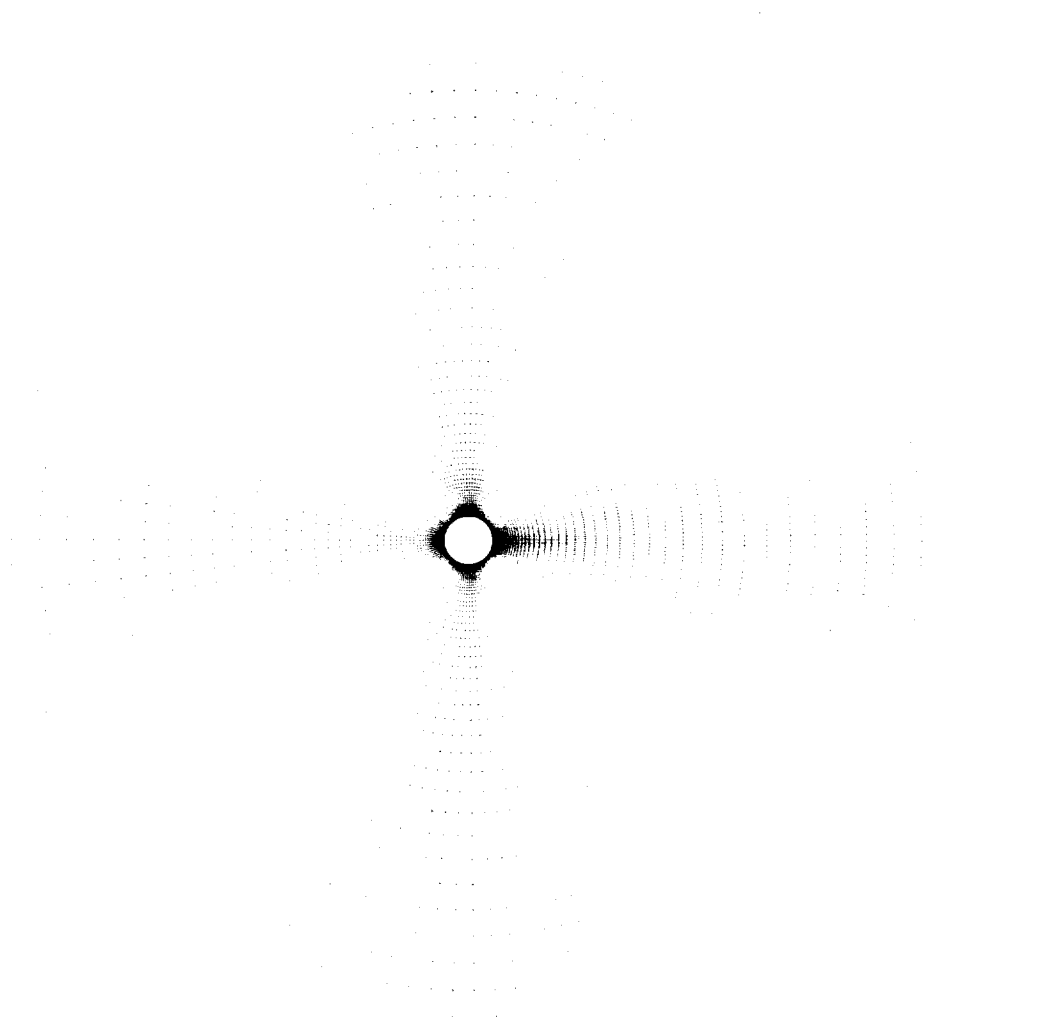


Figure 6.16: 161x65 polar mesh for 2-D vortex shedding behind a circular cylinder in crossflow test case (Reynolds number = 200.0).

Figure 6.16) was used for all calculations. The near cylinder surface normal mesh spacing was 0.001 (normalized by cylinder diameter). The solutions were advanced in time until a sufficient number of shedding cycles had passed (usually 40) to ensure that the solution was periodic in time. For each time increment, calculations were performed using the implicit inner iterations strategy both with and without multigrid. The implicit inner iteration loop was recursively performed until a fixed solution residual convergence criteria was met. Lift and drag coefficient histories are presented for each of the 3 time increment calculations (40, 80, and 160 steps per cycle) for both the non-multigrid and multigrid-based implicit iterative algorithm in Figures 6.17-6.22. In each figure, the pressure and viscous contributions to both the lift and drag coefficients, as well as the total sum for each case are plotted. The initial transient, and ultimate time periodic behavior of the solutions is apparent for each calculation.

For the case utilizing an implicit increment corresponding to 40 time steps per cycle, a second set of calculations was performed with a convergence criteria which required an additional full order of magnitude residual reduction compared to the previous calculations in order to determine the effect of inner iteration convergence level on the predicted results. These calculations are referred to as *highly converged* (HC) results.

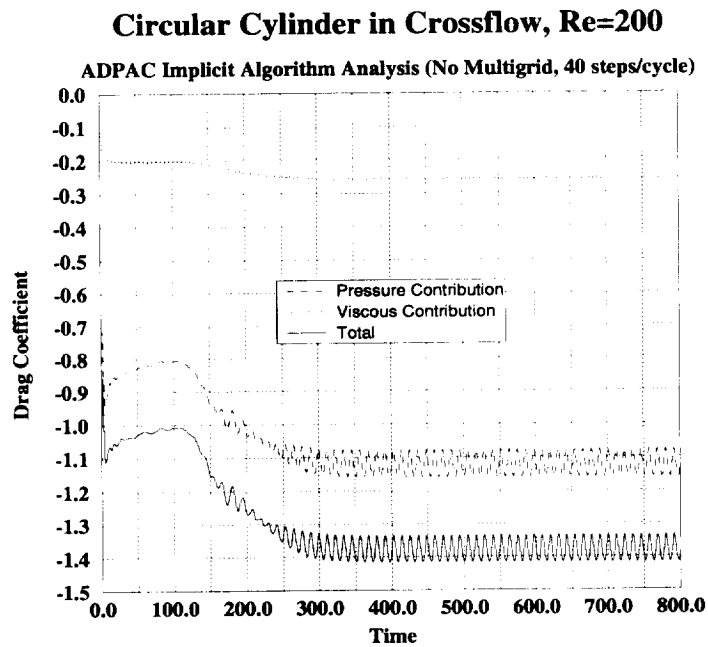
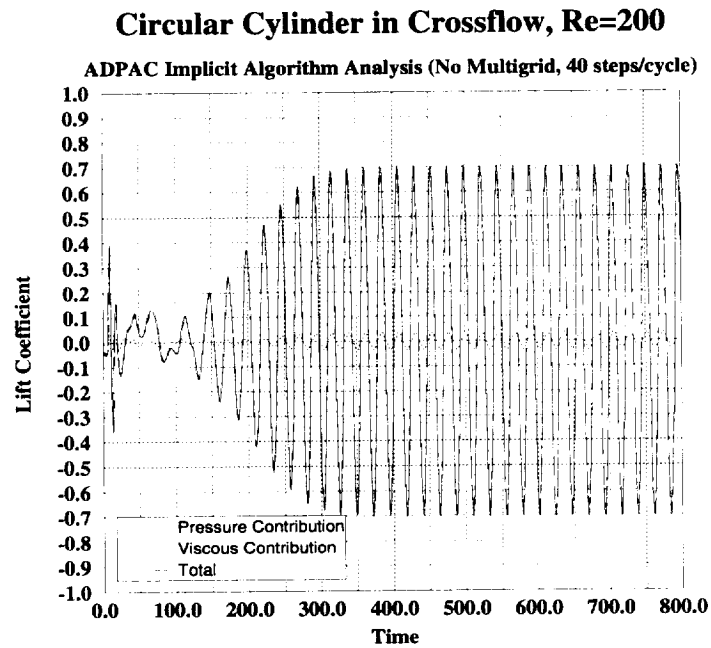


Figure 6.17: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, no multigrid, 40 time steps per oscillation cycle).

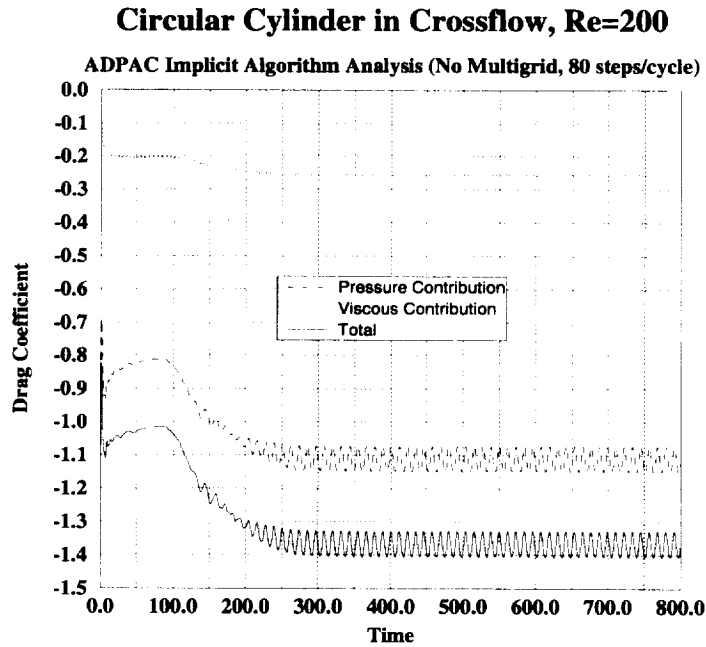
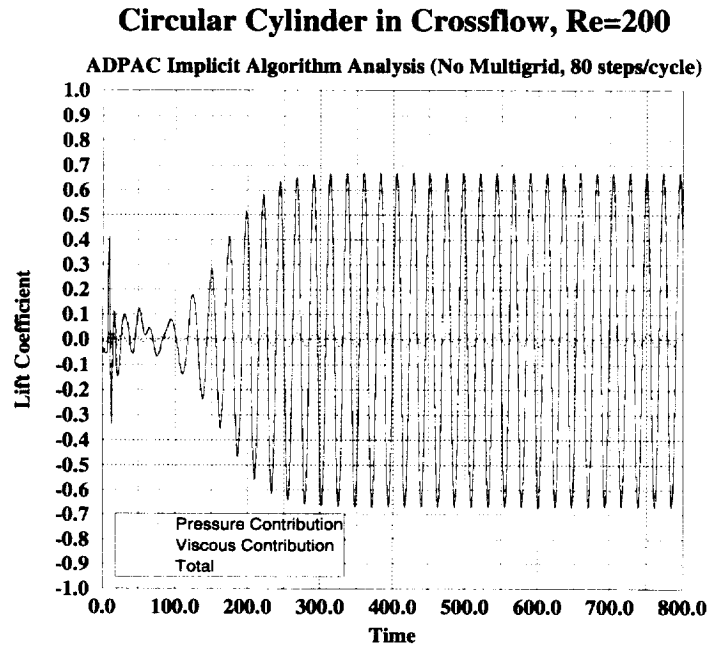


Figure 6.18: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, no multigrid, 80 time steps per oscillation cycle).

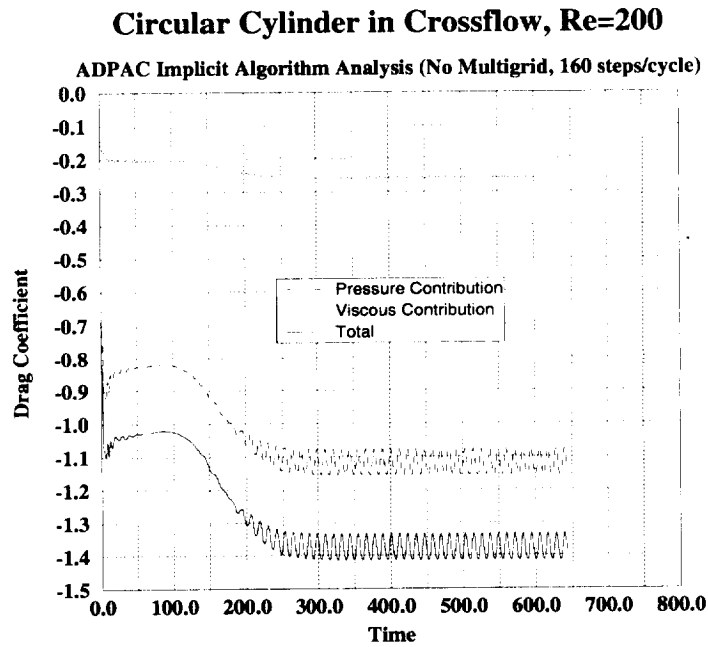
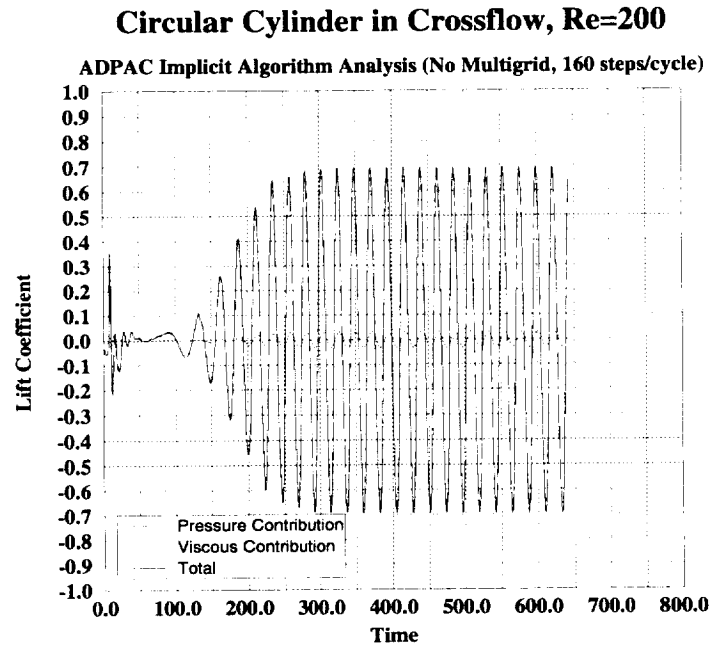


Figure 6.19: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, no multigrid, 160 time steps per oscillation cycle).

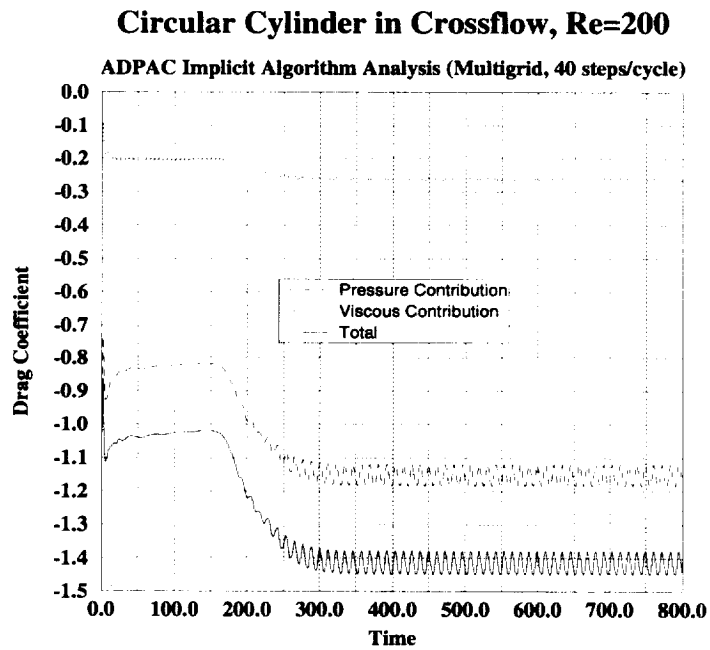
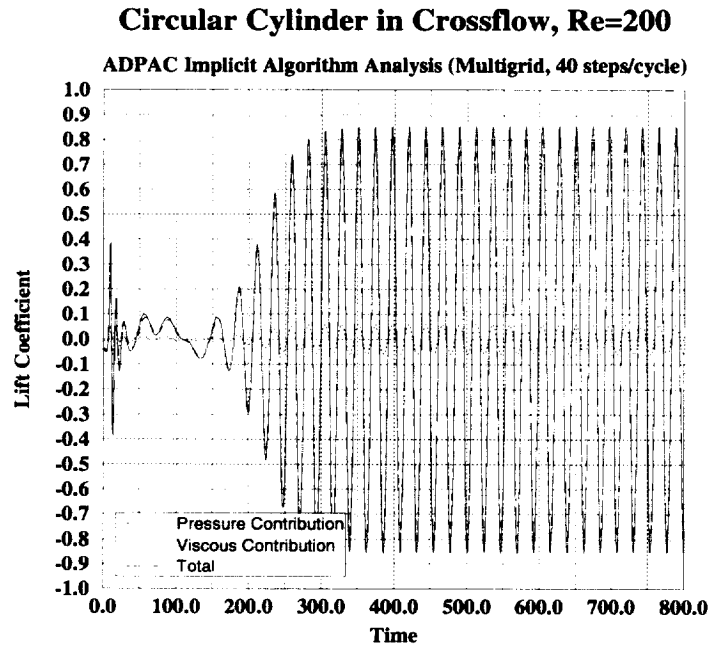


Figure 6.20: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, multigrid, 40 time steps per oscillation cycle).

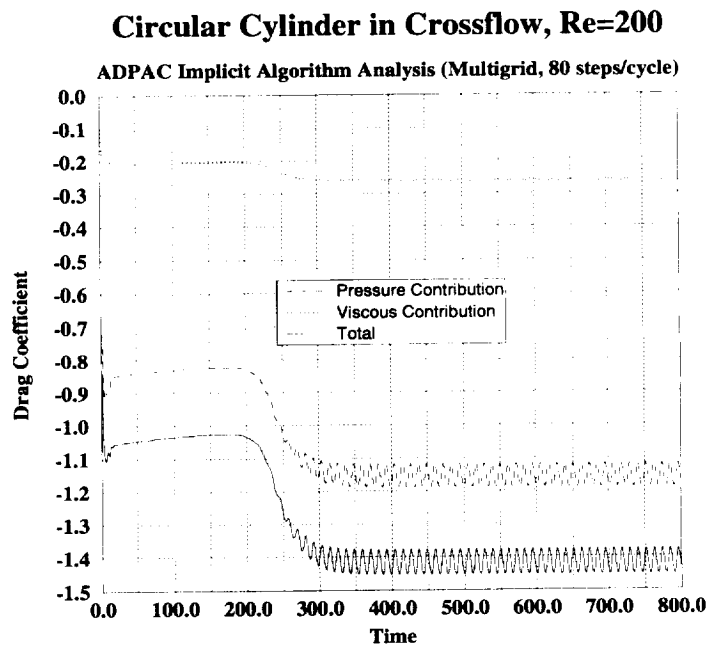
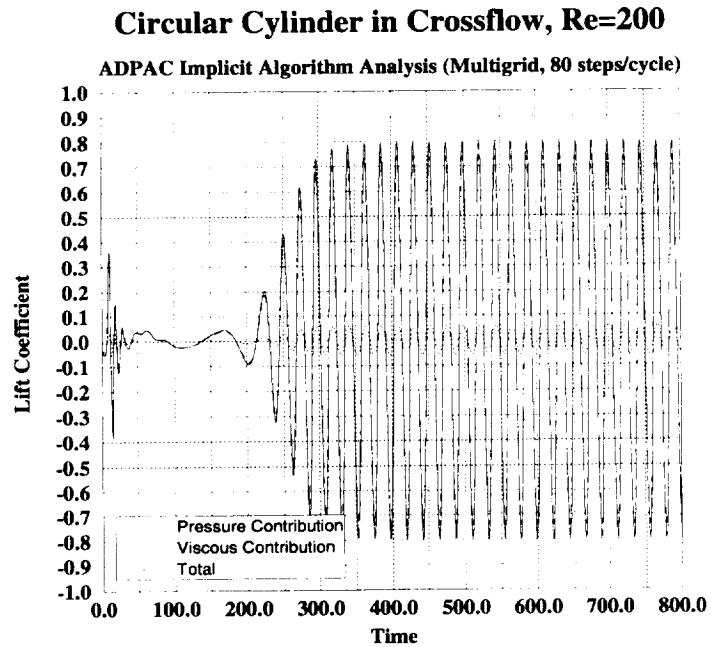


Figure 6.21: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, multigrid, 80 time steps per oscillation cycle).

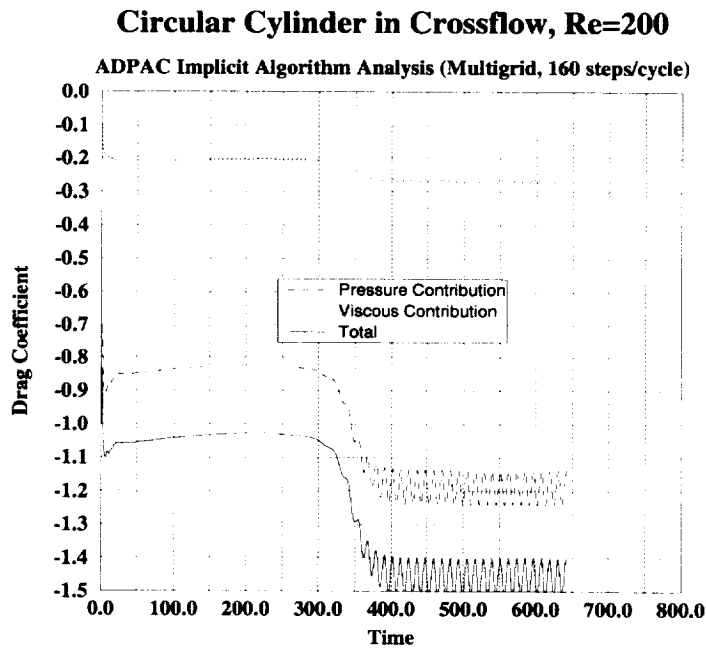
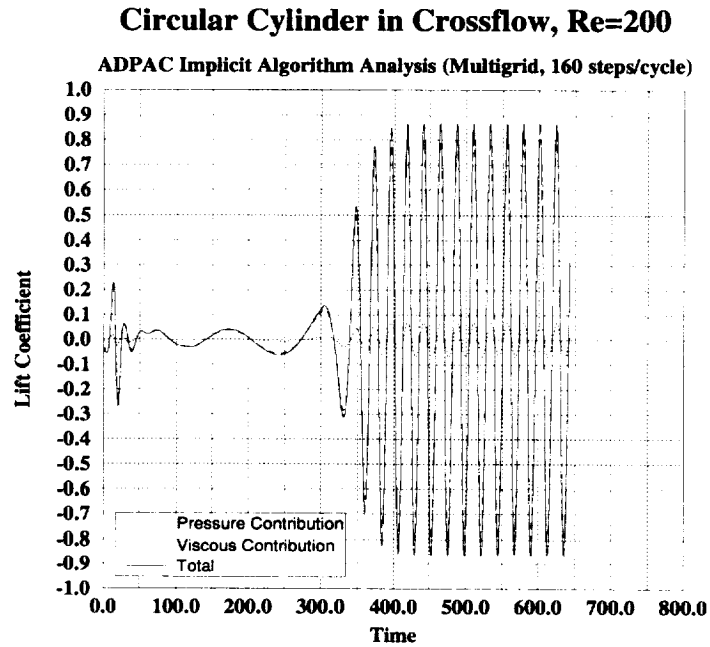
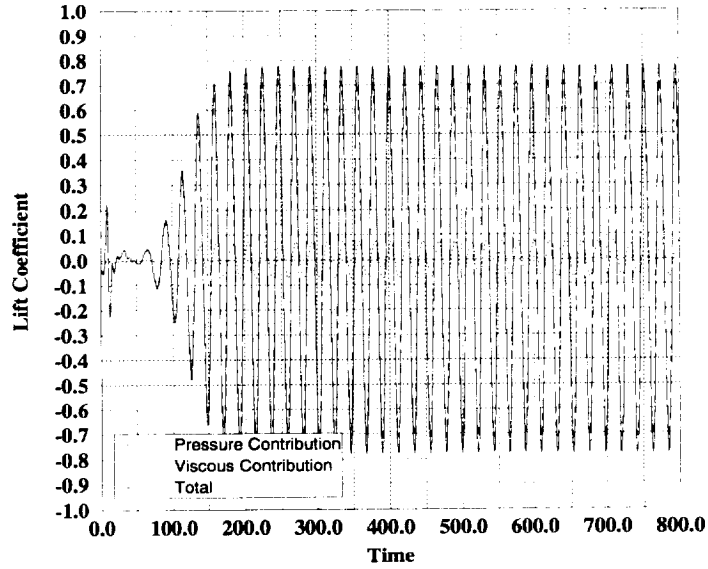


Figure 6.22: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, multigrid, 160 time steps per oscillation cycle).

Circular Cylinder in Crossflow, Re=200

ADPAC Implicit Algorithm Analysis (No Multigrid, 40 steps/cycle, HC)



Circular Cylinder in Crossflow, Re=200

ADPAC Implicit Algorithm Analysis (No Multigrid, 40 steps/cycle, HC)

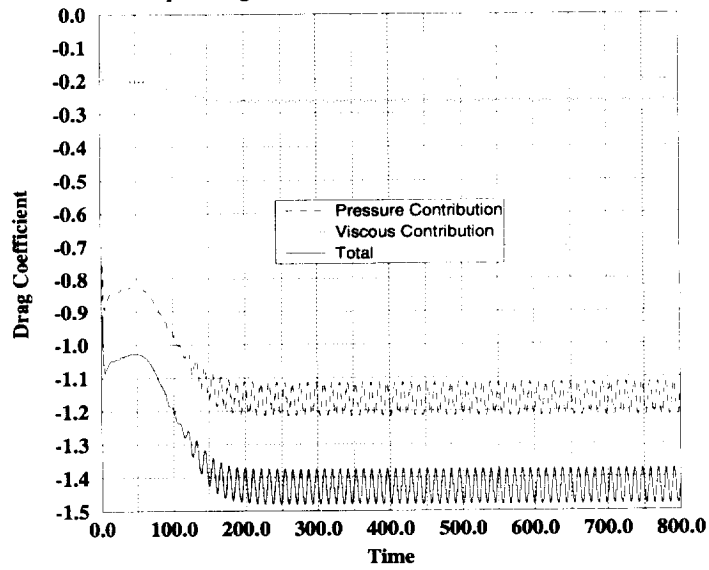


Figure 6.23: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, no multigrid, 40 time steps per oscillation cycle, highly converged).

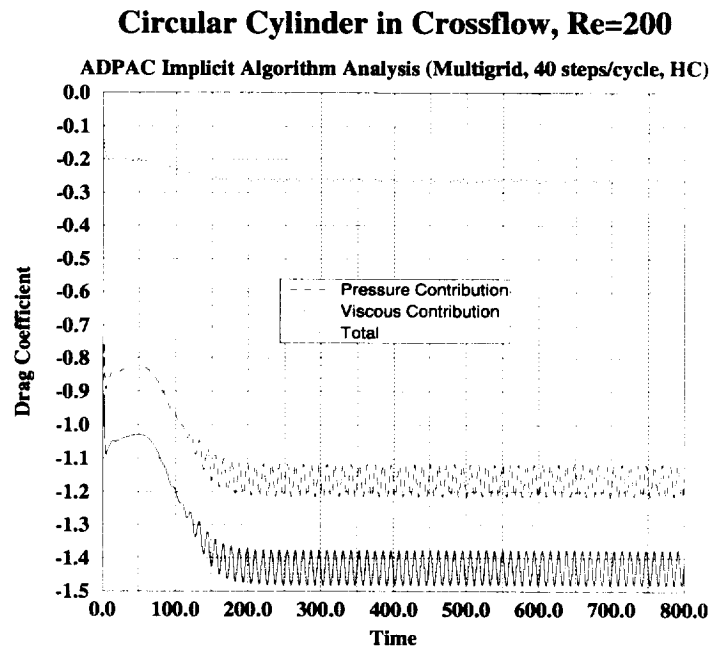
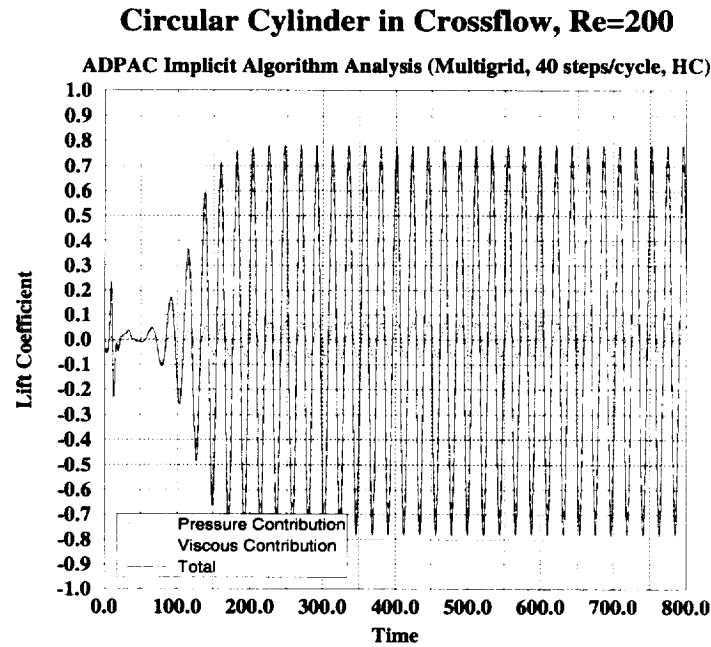


Figure 6.24: Predicted lift and drag coefficient time histories for circular cylinder in crossflow test case (Reynolds number = 200.0, multigrid, 40 time steps per oscillation cycle, highly converged).




Figure 6.25: Predicted instantaneous streamlines for vortex shedding behind a circular cylinder (Reynolds number = 200.0).

A visualization graphic is displayed in Figure 6.25 for the 40 steps/cycle test case with multigrid. This figure illustrates instantaneous streamlines (a steady state based limitation of the plotting package) for the flow at a point in the oscillation history. The formation of the vortex and oscillating flow downstream illustrate that the known aerodynamic structure has been captured correctly.

A summary of integrated results from the present predictions and other published data for this test case is given in Table 6.2. This chart indicates the predicted variations in lift and drag coefficients, as well as the predicted Strouhal number of the vortex shedding cycle. The implicit *ADPAC* predictions were all in very good agreement with experimental data and other recently published data. It should be mentioned that although there is some variation in the *ADPAC* predictions from case to case, this variation is significantly smaller in magnitude than variations which arise from code to code (compare, for example the second order and fourth order results of LeCointe and Piquet [56]). The highly converged cases (labeled with HC) clearly indicate that for a sufficiently converged inner iteration solution, the multigrid and nonmultigrid calculations yield identical results.

A comparison of solution CPU time per oscillation cycle for each calculation is given in Table 6.3. The minimum, maximum, and average CFL numbers for each calculation

Vortex Shedding Off a Circular Cylinder (Reynolds Number = 200)

Summary of Results

		C_D	C_L	St
ADPAC	(No MG, 40 steps/cycle)	1.376 +/- 0.0406	+/- 0.703	0.185
	(No MG, 80 steps/cycle)	1.367 +/- 0.0393	+/- 0.671	0.184
	(No MG, 160 steps/cycle)	1.374 +/- 0.0404	+/- 0.693	0.187
	(No MG, 40 steps/cycle,HC)	1.428 +/- 0.0534	+/- 0.777	0.193
	(MG, 40 steps/cycle)	1.414 +/- 0.0355	+/- 0.853	0.184
	(MG, 80 steps/cycle)	1.415 +/- 0.0383	+/- 0.796	0.189
	(MG, 160 steps/cycle)	1.457 +/- 0.0553	+/- 0.865	0.186
	(MG, 40 steps/cycle,HC)	1.429 +/- 0.0534	+/- 0.782	0.193
Kelecyc (Allison)		1.373 +/- 0.0468	+/- 0.656	0.189
Rosenfeld (<i>Ref. 1</i>)		1.334 +/- 0.0440	+/- 0.676	0.197
Rogers and Kwak	3rd Order	1.29 +/- 0.05	+/- 0.75	0.160
	(<i>Ref. 2</i>) 5th Order	1.23 +/- 0.04	+/- 0.65	0.185
Rosenfeld (<i>Ref. 3</i>)		1.40 +/- 0.04	+/- 0.70	0.201
Lecointe and Piquet	2nd Order	1.46 +/- 0.0400	+/- 0.70	0.227
	(<i>Ref. 4</i>) 4th Order	1.58 +/- 0.0035	+/- 0.50	0.194
Martinez (<i>Ref. 5</i>)		1.27 +/- 0.0035		
Lin et al (<i>Ref. 6</i>)		1.17		
Thoman and Szewczyk (<i>Ref. 7</i>)		1.17 +/- 0.0050		
Wille (experiment) (<i>Ref. 8</i>)		1.3	-----	----
Kovaszny (experiment) (<i>Ref. 9</i>)		----	-----	0.190
Roshko (experiment) (<i>Ref. 10</i>)		----	-----	0.190

Table 6.2: Summary of *ADPAC07* lift and drag coefficient and Strouhal number predictions for vortex shedding behind a circular cylinder in crossflow (Reynolds number = 200.0).

are presented. Estimated results for an equivalent explicit calculation are also provided in this comparison. The utility of using multigrid in the inner iteration cycle appears to be dependent on the maximum CFL numbers of the implicit calculation. This observation was also noted by Arnone [74] in a study of a 2-D rotor/stator interaction problem. The fastest solution time per oscillation cycle was obtained using multigrid with 40 implicit time steps per cycle. However, if the implicit strategy is altered to employ 80 time steps per cycle, then the solution proceeds faster without multigrid. In either case, however, the solution time per cycle is still significantly lower than the previous explicit algorithm. Naturally, these results are dependent on the level of convergence selected for the implicit inner iteration loop. The results for the highly converged test cases suggest that as the convergence criteria becomes more strict, the advantage obtained by the multigrid algorithm may increase. It also seems that although the multigrid and non-multigrid results are converged to the same residual level, the original (less strict convergence level) multigrid results appear to more closely approximate the highly converged predictions than the non-multigrid results. This indicates that residual level alone is probably not a completely accurate measure of inner iteration convergence.

ADPAC Implicit Algorithm CFL and CPU Time Comparison Vortex Shedding Over a Circular Cylinder Test Case

	CFL_{min}	CFL_{avg}	CFL_{max}	CPU/CYCLE (seconds)		Implicit/Explicit Ratios	
				No MG	With MG	CFL	CPU
40 steps/cycle	2.500	14.0	3000.0	512.64	470.59	429	22.8
80 steps/cycle	1.250	7.0	1500.0	484.40	625.06	214	22.2
160 steps/cycle	0.625	3.5	750.0	548.48	824.62	107	19.6
40 steps/cycle, LC	2.500	14.0	3000.0	1680.90	1210.76	429	6.4
Explicit, 17,170 steps/cycle (estimated)			7.0	10735.5	---		

HC – Highly converged test ($\text{Log}_{10}(\text{L2norm}(\text{Sum of Residuals})) < -7.5$)
 All other tests ($\text{Log}_{10}(\text{L2norm}(\text{Sum of Residuals})) < -6.5$)
 MG – Multigrid (3 levels, no subiterations)

161x65 O-type mesh with 0.001 x diameter near surface normal mesh spacing. Outer boundary was placed at 10.0 diameters from cylinder center.

All calculations initiated from a steady state solution at $Re=50$ and 3.0 degrees angle of attack to initiate asymmetry.

CPU times are for a Silicon Graphics PowerChallenge, Operating System IRIX 6.0, compiler level 6.0, compiled with "-O3 -mips4 -WK -c=0 -so=2 -ro=0 -OPT:roundoff=3 -OPT:IEEE_arithmetic=3 -lfastm" compiler options.

Table 6.3: Summary of *ADPAC07* CPU times for vortex shedding behind a circular cylinder in crossflow (Reynolds number = 200.0).

6.6 Penn State Research Compressor Rotor/Stator/Rotor Interaction

The purpose of this test cases was primarily to examine the stability of the modified (more "implicit") iterative multigrid-based implicit algorithm. This test case had previously exhibited numerical instabilities and was therefore an indicator of the known problems for the original (Arnone [73]-based) implicit algorithm.

The geometric model is based on a simplified rotor/stator/rotor interaction model originally developed and utilized for an Allison study of the Penn State research compressor. This mesh is rather unique in that a body-centered O-type mesh is employed locally about each blade, and several H-type meshes are then used to discretize the remainder of the blade passage. This mesh is illustrated in Figure 6.26. The resulting mesh provides excellent resolution in near blade surface velocity profiles, and provides a mechanism for accurately tracking wakes in the connecting H-grids which is not available in an O-type mesh alone. The original blade counts for the rotor/stator/rotor combination (72, 73, 74) were modified (73,73,73) to permit a 1:1 ratio between rotors and stator. Therefore only a single blade passage representation per blade row was required for the analysis. The implicit algorithm was applied to this flow in an attempt to predict the time-dependent aerodynamic interactions resulting from the rotor/stator/rotor relative motion. The solution was initiated from an existing *ADPAC* restart file. The time step was chosen such that 50 global implicit iterations were utilized during each blade passing period. This resulted in an equivalent CFL number over 10,000. A total of 15 inner iterations were utilized during each global implicit iteration.

**Penn State Research Compressor
ADPAC Rotor/Stator/Rotor Aerodynamic Interaction
Hybrid O-H Mesh System**

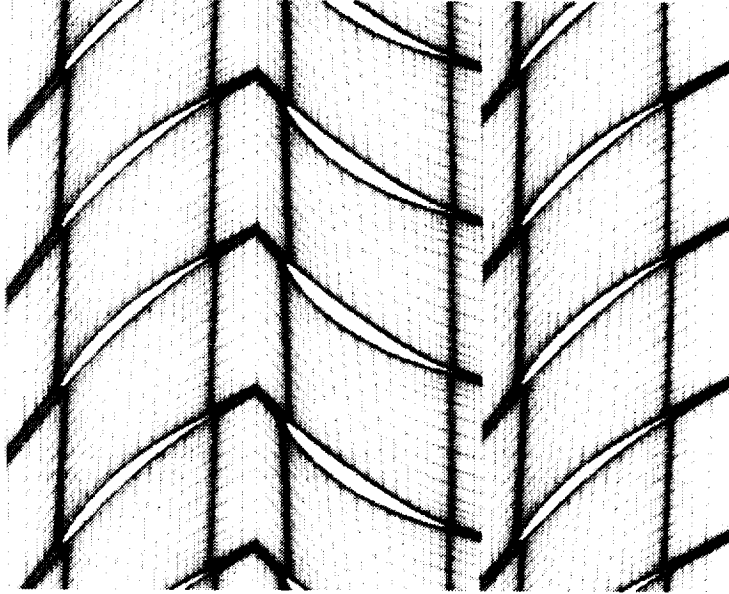


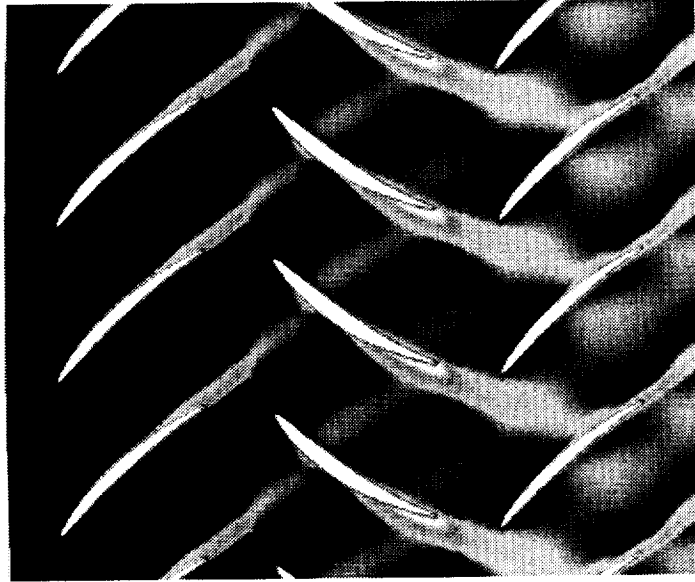
Figure 6.26: Hybrid O-H mesh system for Penn State research compressor rotor/stator/rotor interaction analysis.

The original implicit algorithm based on the “explicit” iteration strategy described by Arnone [73] was previously shown to develop a numerical instability for this test case. Attempts to circumvent this instability via time step restrictions were generally unsuccessful, and this failure essentially prompted the search for a more stable, less restrictive algorithm. Application of the modified algorithm with “implicit-like” inner iterations demonstrated superior stability, and a stable time-marching process was demonstrated. The solution was ultimately advanced to a time-periodic state. A sample graphic from the time-periodic solution is given in Figure 6.27. This figure illustrates instantaneous entropy contours at both a midspan and a near tip radial plane. The propagation of the rotor wakes into the stator blade row, and subsequent transport into the downstream rotor are clearly captured.

6.7 Advanced Small Turboshift Compressor (ASTC) Wall Function Demonstration Calculation

Numerical calculations based on the *ADPAC07* algebraic (Baldwin-Lomax) turbulence model with wall functions are described below to demonstrate the effectiveness of this scheme. A series of predictions from the *ADPAC07* analysis using wall functions was compared with results from the average-passage analysis [32] for the Advanced Small Turboshift Compressor (ASTC) geometry. This turbomachine is a small 2-stage high-performance compressor

Near Midspan



Near Tip

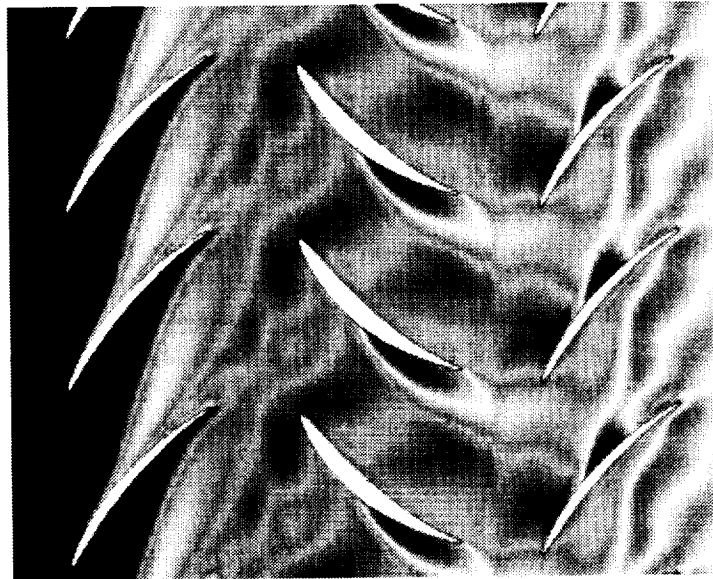


Figure 6.27: Penn State research compressor rotor/stator/rotor interaction analysis - predicted instantaneous entropy contours.

Prediction Scheme	Predicted Mass Flow / Design	Predicted Pressure Ratio/ Design
<i>ADPAC</i>	<i>1.032</i>	<i>1.0</i>
<i>Average Passage Analysis</i>	<i>1.0267</i>	<i>1.002</i>

Table 6.4: Comparison of predicted performance figures for ASTC compressor.

which was tested at NASA-Lewis Research Center. The average-passage analysis results were taken from an existing solution generated during the design of the compressor. The *ADPAC07* analysis was applied to this flow by separating the average-passage analysis mesh systems into blade local meshes, and applying the circumferential averaging interblade row coupling technique (mixing planes) between mesh sections. Small differences between the two calculations therefore arise due to the inherent losses generated by the mixing plane coupling scheme and the effect the placement of the mixing plane has on the blade local flow. In spite of these differences, the *ADPAC07* solution was found to be in excellent agreement with the average-passage analysis [32] for this flow, as demonstrated by the comparison of predicted performance parameters given in Table 6.4. The overall performance predictions from both codes were found to be in excellent agreement. This agreement is not completely unexpected as the *ADPAC07* code and the average-passage code utilize similar numerics and wall functions. An illustration of the predicted surface static pressure contours for the compressor at the design condition is given in Figure 6.28. This figure serves to illustrate the magnitude of this multistage calculation, and indicates the usefulness of this approach for complex turbomachinery flow predictions.

6.8 NASA Rotor 67 Serial/Parallel Comparison

To verify the operation of the parallelized code, an Euler solution for NASA Rotor 67 was run both in serial, and in parallel with 8 processors on an nCUBE 2 parallel computer. The serial version of *ADPAC* (prior to parallelization) was run on an IBM RS6000 workstation. Both solutions used 8 blocks with a total grid size of 49x17x17. Figure 6.29 shows the resulting static pressure contours on the suction surface of the airfoil, and on a blade-to-blade plane at 84% span. Although the plots show the same solution, there are differences

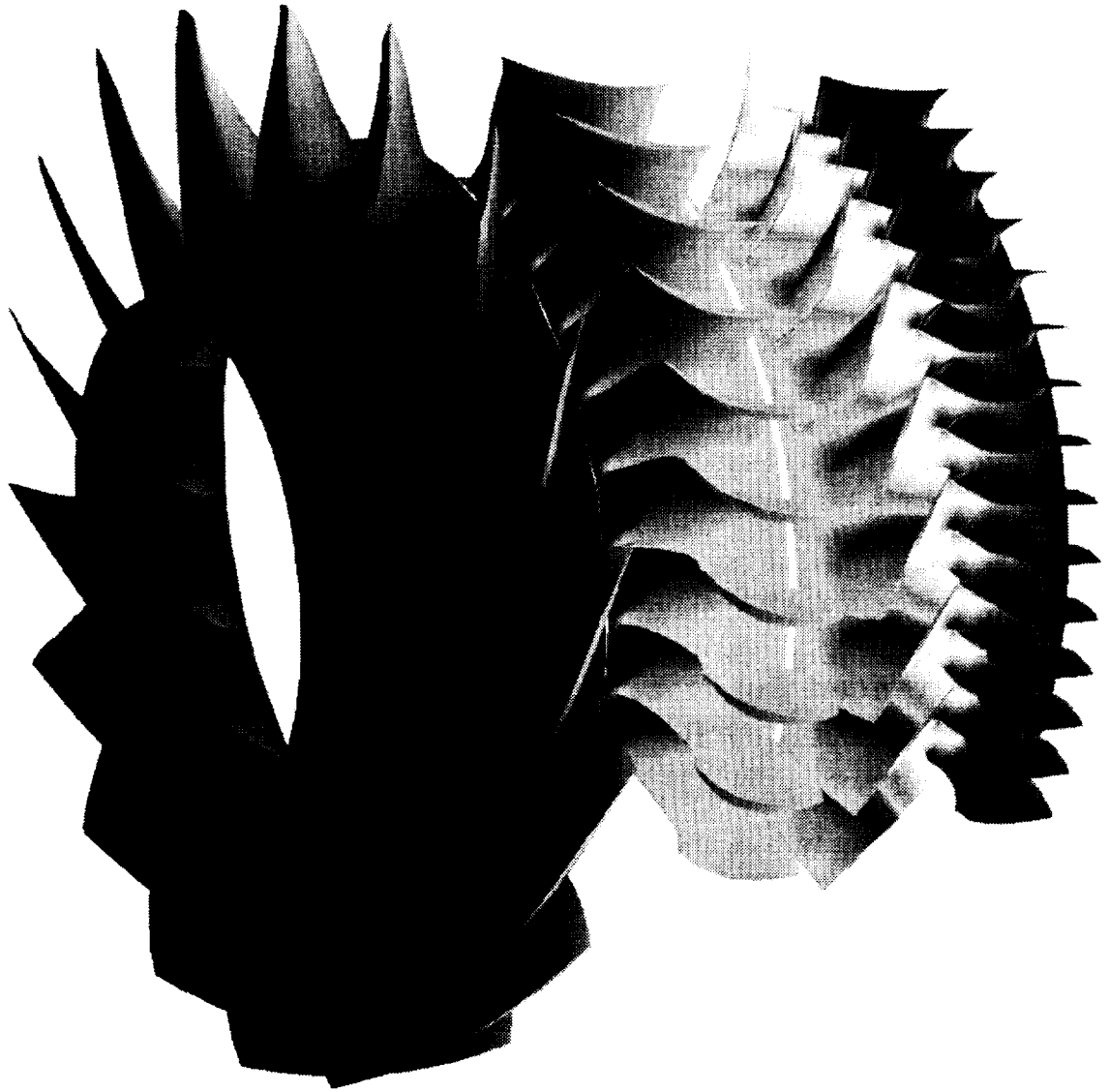


Figure 6.28: *ADPAC'07* predicted surface static pressure contours for ASTC' compressor.

between the binary restart files, probably caused by different rounding schemes on the two computers.

A solution for Rotor 67 using only one block (with the same grid as the 8 block case) was also run on the IBM workstation. Plots from these solutions are shown in Figure 6.30. The minor differences between the solutions arise from two sources: extrapolation of information to the vertices at the block edges (converting cell-centered data to vertex centered data for plotting purposes), and small differences resulting from the fact that dissipation quantities were not passed across block boundaries for the 8 block calculation (the *ADPAC07* code was later modified to permit this capability). These items are explained more completely in the paragraphs which follow.

The values at the edges of the blocks (phantom cells) are extrapolated from neighboring cells. This results in quantities which are not the same as those from the single block run, where these points are in the interior of the solution. The latest release of *ADPAC* employs a better extrapolation scheme, which reduces the problem. It is also possible to make changes to the patching algorithm which would eliminate the extrapolation by loading the corner phantom cells with the proper values from the neighboring block.

The dissipation (damping) terms were not passed as part of a patched boundary in this calculation. The dissipation routines set the dissipation fluxes at all block boundaries to zero, which ensured global continuity within each block. This also means that the multiple block run had different dissipation values than the single block run at the interior boundaries. The dissipation routines were ultimately modified to allow patched boundaries to share dissipation information.

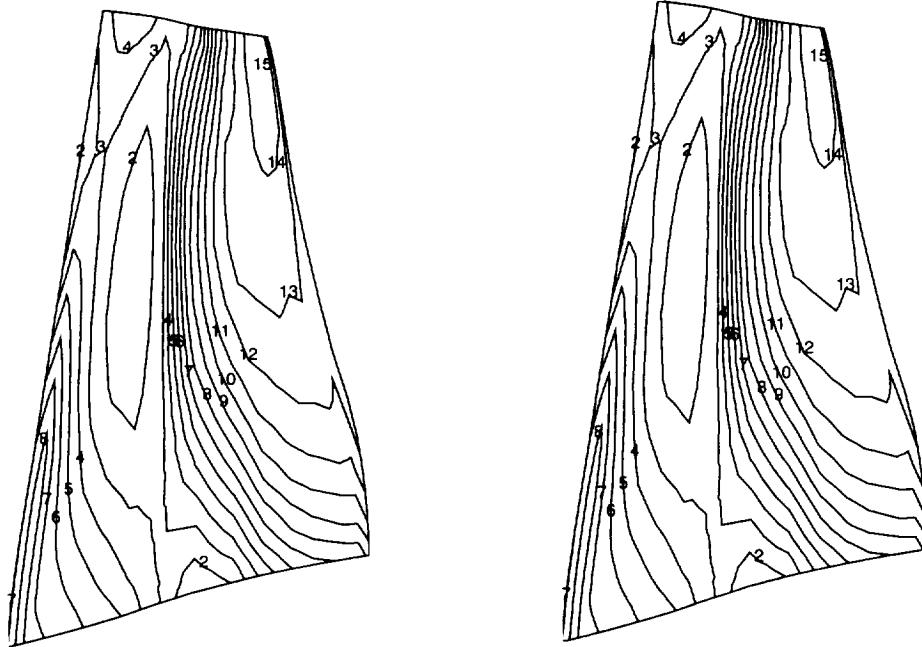
Neither of these problems is directly related to parallelization, but arise from the use of multiple blocks. Therefore the test of the parallel code was considered successful when the results matched a multiple block run on a serial machine.

6.9 Advanced Small Turboshaft Compressor (ASTC) Parallel *ADPAC07* Rotor/Stator Aerodynamic Interaction Analysis

The Advanced Small Turboshaft Compressor (ASTC) is a high performance, high speed two stage compressor design being studied jointly by Allison, NASA, and the U.S. Army. The *ADPAC07* code was used extensively as a design analysis tool under the ASTC program, and as part of the design analysis, a time-dependent aerodynamic analysis of the two stage rotor/stator/rotor/stator aerodynamic interactions was performed. This problem was particularly well suited for analysis using the parallel computing features of the *ADPAC07* code since it involves large computational run times, many mesh points, and a large number of mesh blocks. The ASTC case was therefore selected as a demonstration and evaluation test case for the parallel computing features of the *ADPAC07* code for this study. The ASTC was analyzed in time-dependent mode on the NASA-Lewis LACE (RS/6000) cluster of workstations. Since the simulation involves multiple blade rows, each blade passage was represented by a single H-type mesh constructed algebraically using the code developed by Mulac [81]. The original compressor blade counts were modified from the design values of 16, 36, 33, and 60 (rotor #1, stator #1, rotor #2, stator #2) to 16, 36, 32, and 60 to facilitate a reduction of the problem size to one quarter of the wheel. The configuration requires 36 blocks to obtain a periodic geometry for each of the four blade rows using one block per airfoil passage. The total mesh size was about 2,000,000 grid points, or about

Serial ADPAC vs. Parallel ADPAC

Rotor 67 Solution, 49x17x17 Grid



Suction Surface Static Pressure Contours



Static Pressure Contours at 84% Span

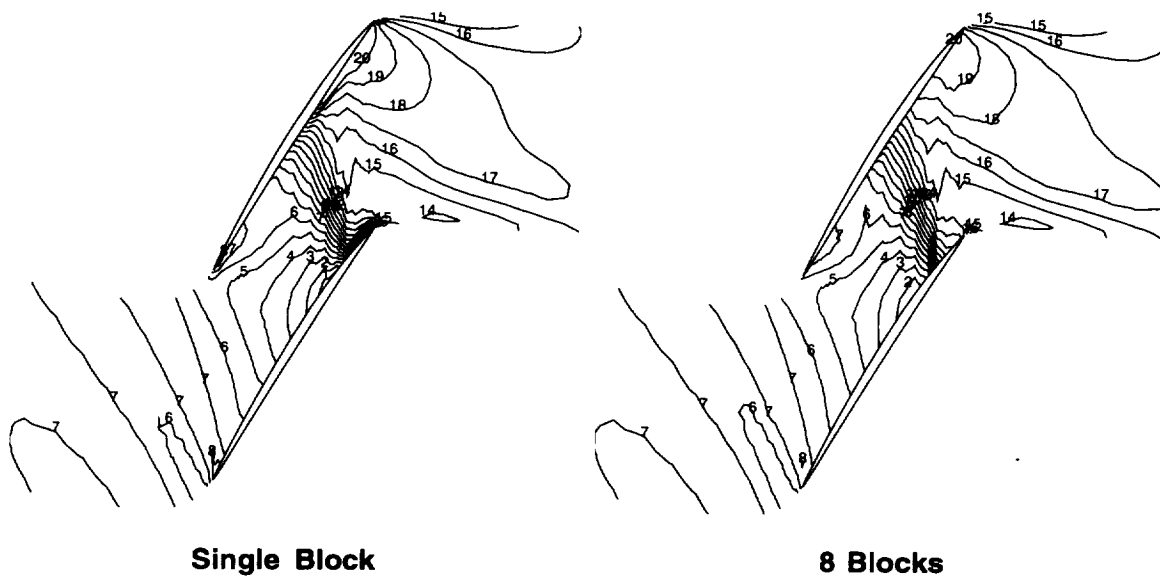
Serial

Parallel

Figure 6.29: Serial and parallel solutions of Rotor 67 converge to the same answer.

Single vs. Multiple Block ADPAC Solution

Rotor 67 Solution, 49x17x17 Grid



Static Pressure Contours at 84% Span

Figure 6.30: Single block and multiple block solutions for Rotor 67, run on a serial computer, differ slightly.

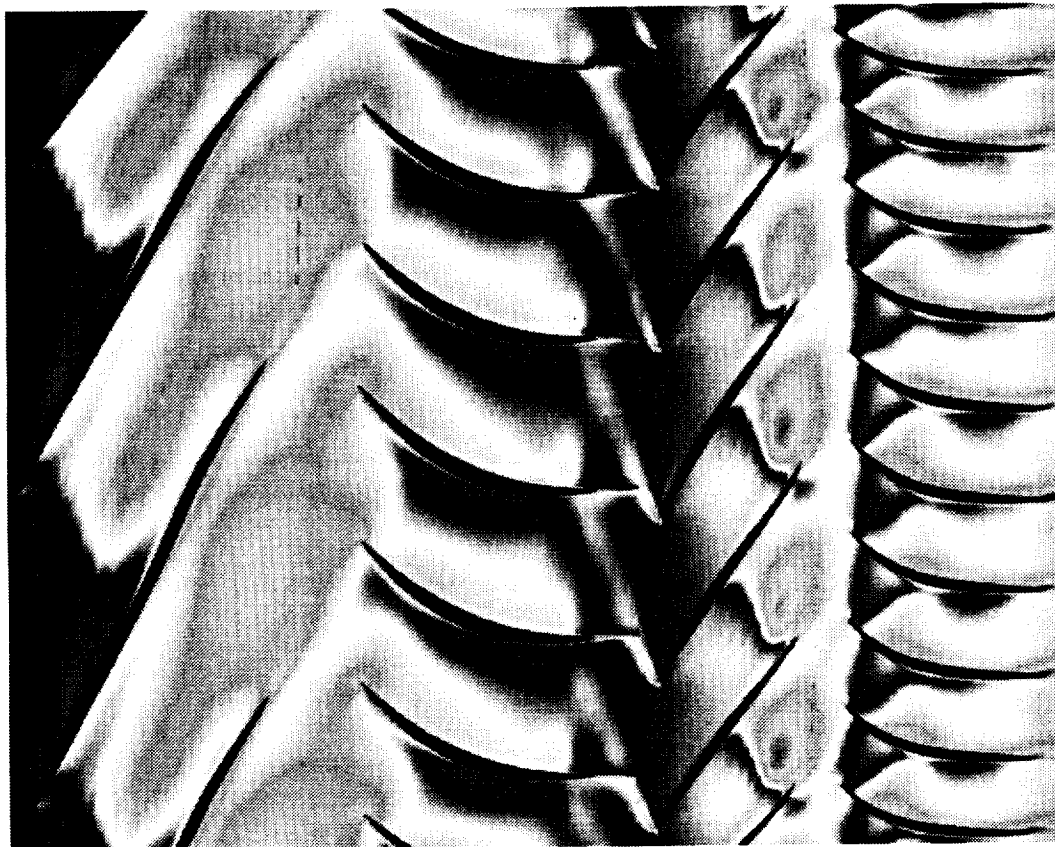


Figure 6.31: Predicted velocity magnitude contours for ASTC compressor *ADPAC07* rotor/stator aerodynamic interaction analysis.

55,600 points per blade passage. The calculation was advanced using the direct explicit Runge-Kutta time-marching algorithm in the *ADPAC07* code, as the implicit algorithm was not yet developed at the time of this analysis. Consequently, the analysis progressed rather slowly, but a reasonable solution eventually evolved. The time step requirement for time accuracy and stability was very small, which meant that many steps (roughly 10,000) were needed to advance the solution by one period. An illustration of the instantaneous predicted velocity magnitude field is given in Figure 6.31. The time-dependence of the flow is illustrated by the variance in contours between blocks of the same row (e.g. the shock position in the 2nd stage rotor).

Code performance data was collected during the ASTC runs in the form of elapsed wall times. Since the performance testing was done in a dedicated environment, (all machines except the server were dedicated), these times should mirror CPU times. Timing runs were made both with and without the ALLNODE switch, which is a special hardware device which enables low latency communications between clustered IBM RS6000 workstations. Parallel performance is highly problem and machine dependent, but results should be representative of large turbomachinery simulations. All simulations were made using 9 IBM RS6000 model 560 workstations, each loaded with 4 airfoil passage blocks per processor. In this case, two boundary conditions require communications: **BCPRRM** (inter-blade row rotor/stator interaction boundaries) and **PATCH** (direct inter-block communication (periodic) boundaries). (Additional details on boundary names and procedures are available

in the *ADPAC07* User's Manual [28].) Of these, most of the communicating is done in **BCPRRM**, because many PATCH boundaries are between blocks on the same processor.

Unfortunately, it was not possible to isolate the time associated with communications because the communication calls were embedded throughout the code. However, the amount of time spent in applying boundary conditions can be determined, and a substantial portion of this time is due to communications. The total time spent advancing the solution is also determined illustrating the relative importance of communications.

Running without the ALLNODE switch, the total time per iteration was 382 seconds, or about 9.4 steps per hour. Of this, the application of boundary conditions was 268 seconds which is 70% of the total. Using the ALLNODE switch, the total time per iteration was 217 seconds, or about 16.7 steps per hour. Of this, the application of boundary conditions was 103 seconds which is 47% of the total. The use of the ALLNODE switch improved the speed of the code by a factor of 1.77. In all, this study demonstrates the applicability of the workstation cluster concept for solving very large computational tasks in an efficient fashion.

Chapter 7

STEADY STATE CASING TREATMENT ANALYSIS WITH NO INLET DISTORTION

In this section, several steady state predictions of turbomachinery blade row flowfields both with and without casing treatment are presented. For all of the results presented here, the inlet flow was assumed to be distortion free, and hence, these results are intended to reflect the general differences in blade row aerodynamics when casing treatments are present under essentially ideal flow conditions. The results in this section were derived under NASA Contract NAS3-25270, Task 6 - Casing Treatment Analysis, but are reported formally under the present task.

7.1 MIT Stator Vane with Axial Skewed Groove Casing Treatment

Initial calculations to verify the accuracy of the endwall treatment time-averaged boundary condition developed for discrete endwall treatments were based on experimental results taken on the MIT Compressor Research Rig [7]. A schematic of the MIT Compressor Research Rig is given in Figure 7.1. The hardware tests incorporated a single stage compressor with a cantilever-mounted stator and a rotating hub. Stator hub clearance as a fraction of span was 1.5%. Endwall treatments were incorporated within the rotating hub to facilitate measurements in the blade-relative frame without the use of rotating machinery (as would be required for a rotor). The endwall treatments employed in the experimental studies were square-edged axial skewed slots, as illustrated in Figure 7.2, extending from roughly 5% to 95% axial chord beneath the stator blade row. A total of 170 treatment slots were utilized for the 45 blade stator array. This configuration has been the focus of several extensive experimental [7] and numerical [14] investigations. Experimental data were taken for both a smooth (non-treated hub) and for the hub utilizing the endwall treatments.

The numerical analysis of the MIT stator was performed using a mesh system comprised of 3 blocks as shown in Figures 7.3 and 7.4. H-type meshes were employed for each region of interest in the calculation. Separate mesh systems were incorporated in the blade hub clearance region and the treatment groove to avoid excessive mesh shear normally encountered with H-type meshes for cantilever mounted blades (this implies that the blade passage

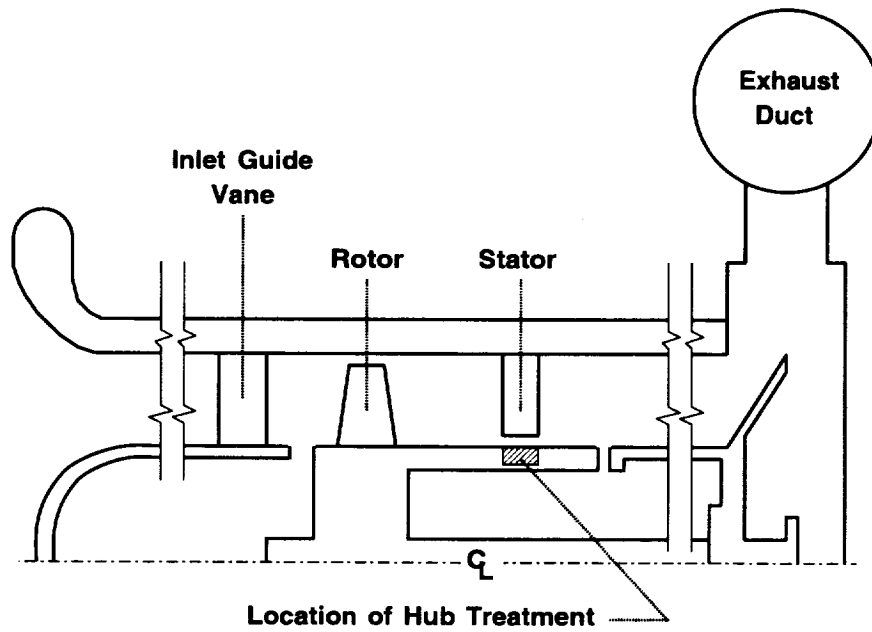


Figure 7.1: MIT compressor research rig illustrating stator hub treatment geometry.

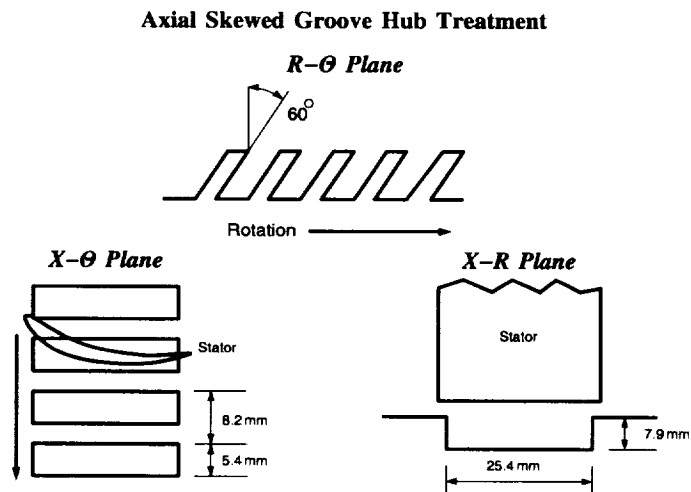


Figure 7.2: MIT stator and axial skewed slot endwall treatment geometry.

mesh was generated as if the blade extended full span, and then the clearance mesh was added to discretize the clearance region). It should be noted that the mesh coordinates at the boundary interfaces between the blade grid and the treatment grid, and the clearance grid and the treatment grid have common x, r coordinates. This axisymmetric coincidence simplifies the boundary communication scheme (circumferential averages may be computed along mesh lines). A direct circumferential average of flow data in the blade passage and clearance meshes defines the boundary specification for the treatment mesh at the interface along the hub. The endwall treatment time-average boundary condition (described in Chapter 5) is employed to complete the boundary specification for the blade passage and clearance meshes along the hub in the region spanned (axially) by the treatment slot. The mesh block sizes were 129x61x33 (axial, radial, circumferential for the blade passage mesh, block 1 in Figures 7.3, 7.4), 65x13x5 (clearance region mesh, block 2 in Figures 7.3, 7.4), and 49x33x33 (treatment slot mesh, block 3 in Figures 7.3, 7.4).

The operating points for these calculations were defined by a specified inlet profile taken from the experimental measurements described in Reference [7]. The exit flow conditions were obtained iteratively in an attempt to match the experimental pressure rise coefficient. Due to the expense of iterating on the 3-D solution, only 3 different operating points were run and the nearest solution was selected for further study. Predictions were obtained for flows both with and without the treatment present and compared with experimental data.

A comparison of the predicted and experimental inlet and exit spanwise total pressure coefficient profiles for the non-treated (smooth hub) and treated (endwall treatment hub) are given in Figures 7.5, and 7.6, respectively. For the untreated hub calculations, the decrease in total pressure coefficient was generally under predicted, although this was, in part, due to the previously mentioned difficulties in matching the exact experimental flow condition. For the comparison of the treated endwall test case, the experimental data displays a *rise* in total pressure near the treated endwall. This rise is believed to be due to the added work afforded to the fluid by the rotating treatment passage. Unfortunately, at best, the predicted data shows a reduced total pressure loss in the outer endwall boundary layer.

Illustrations of the predicted near endwall flowfield for the smooth hub and the treated hub are given in Figure 7.7 and 7.8, respectively. The dark traces represent the trajectories of the clearance flow, while the lighter traces represent the flow entering the blade passage from the treatment. Although not shown here, flow visualization of the predicted near endwall velocity vectors for both the treated and smooth hub solutions demonstrated good qualitative agreement with the experimental data. The effect of the treatment slots appears to be that of sweeping the clearance flow farther aft than that which occurs in the untreated stator. This effect helps to prohibit the buildup of “blockage” resulting from the low momentum, low total pressure, clearance flow vortex in the forward portion of the blade passage. This buildup of blockage is believed to be a significant factor contributing to stall in axial compressor rotors [37].

The presence of the treatment passage acts to recirculate flow near the endwall from the rear of the passage to the front. The percentage of the blade passage flow passing through the treatments was predicted to be approximately 0.5% (compared to 2% estimated from the experimental data). Crook’s results [14] for the MIT stator were based on an imposed treatment flow. From those results, it appears that if the treatment flow is accurately modeled, then the overall blade passage performance can be captured. The present inability to accurately match the experimental data suggests that the endwall treatment time-average boundary condition is somehow not representative of the time-averaged flowfield, or that the exact physical processes associated with the endwall treatments are time-dependent in

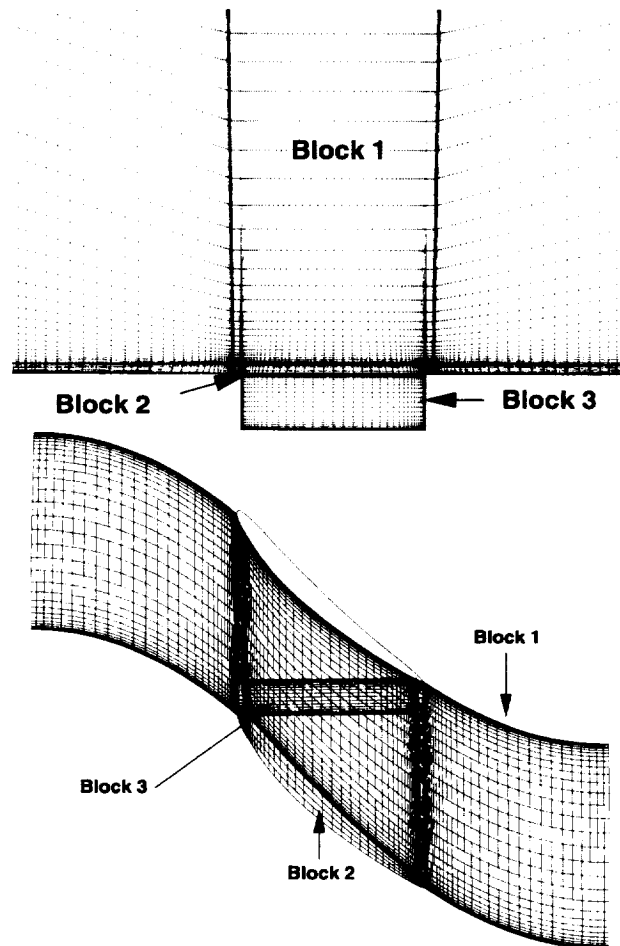


Figure 7.3: MIT stator multiple block mesh system.

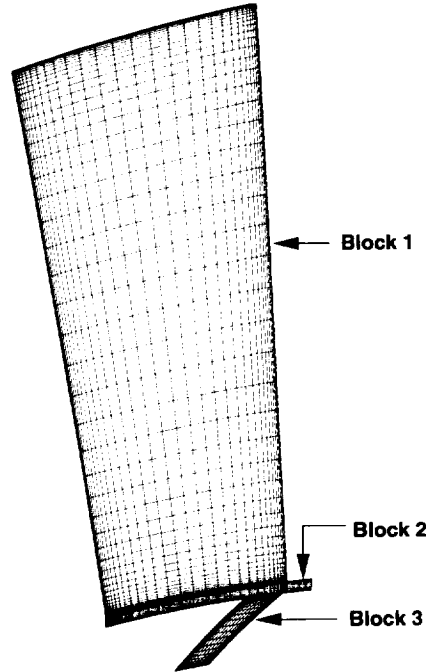


Figure 7.4: MIT stator multiple block mesh system illustrating skewed axial slot endwall treatment mesh.

nature.

The conclusions regarding this test case suggest that for this rather low speed flow, the qualitative characteristics of the treatment flow were correctly predicted, but that the overall magnitude of the flow and the pressure rise through the treatment were somewhat under predicted. Uncertainty in the experimental operating point and in the ability to measure details of the treatment flow (e.g. the percentage of flow which passes through the treatment) for this configuration must also be suspect.

7.2 NASA Rotor 5 with Axial Skewed Groove Casing Treatment

The *ADPAC* analysis was subsequently applied to the NASA Rotor 5 geometry, a high speed, high pressure ratio fan employing 47 blades and a part span shroud. A description of the NASA Rotor 5 geometry, and a summary of the design performance parameters for this fan are given in Figure 7.9.

Several experimental test data sets were available for NASA Rotor 5 utilizing various casing treatments [1], [2]. For the purposes of this paper, the skewed axial slot casing treatment illustrated in Figure 7.10 and described in Reference [1] was selected for study. The results for the treated rotor presented in this section were all obtained using the endwall treatment time-average boundary condition described in Chapter 5.

Constant speed performance curves were generated using the *ADPAC* analysis for NASA Rotor 5 both with and without endwall treatment following the procedure previously outlined. The numerical solutions employed the H-type mesh systems illustrated in Figures 7.11 and 7.12. In this case, two mesh blocks were required for the blade passage due to the

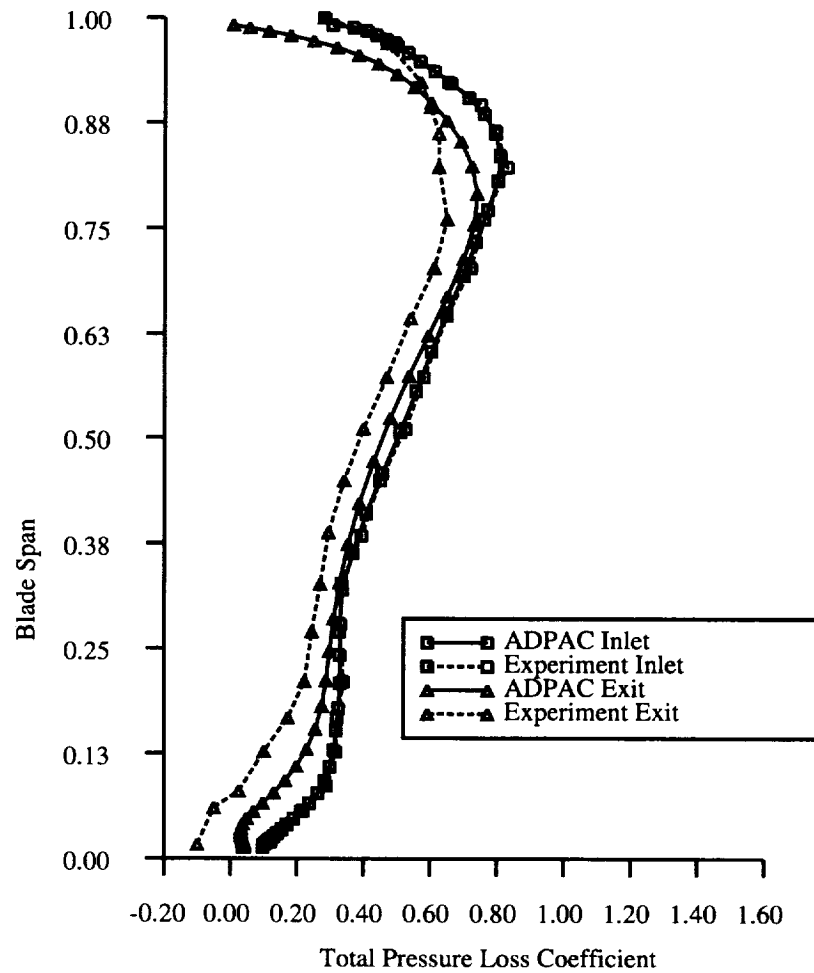


Figure 7.5: Comparison of predicted and experimental inlet and exit radial total pressure coefficient profiles for non-treated MIT stator.

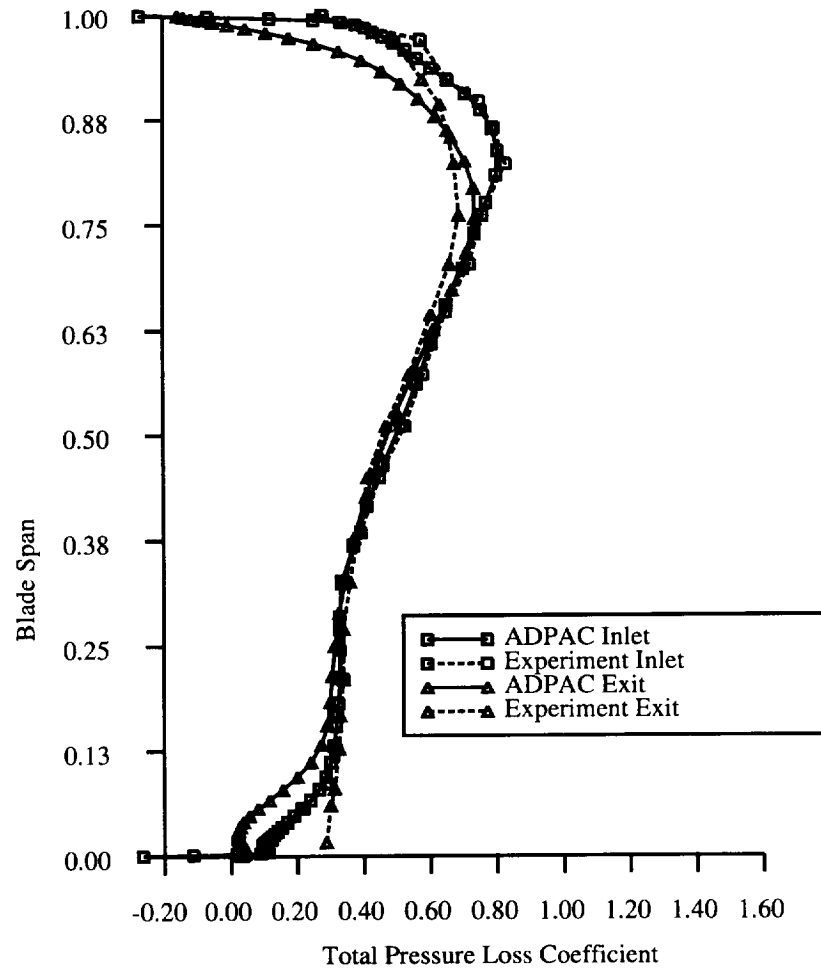


Figure 7.6: Comparison of predicted and experimental inlet and exit radial total pressure coefficient profiles for treated MIT stator.

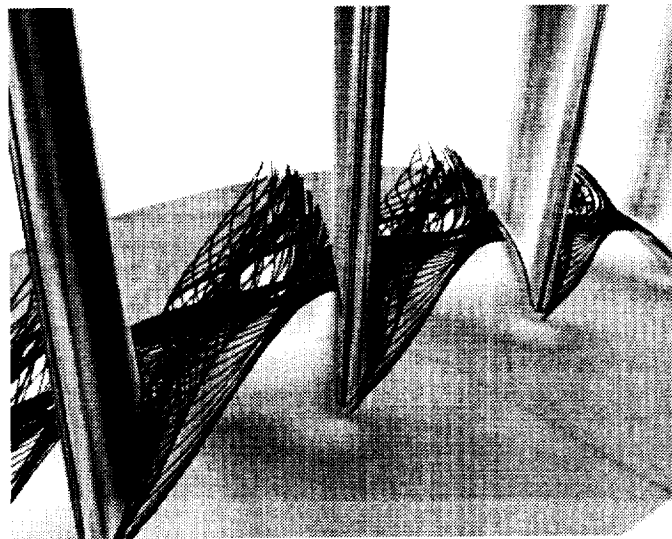


Figure 7.7: Predicted flow visualization of MIT stator with smooth endwall.

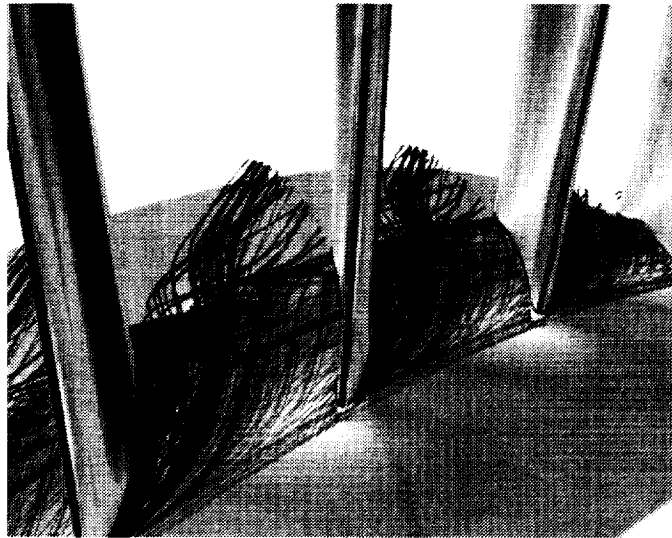


Figure 7.8: Predicted flow visualization of MIT stator with treated endwall.

Rotor 5 Verification Study Description

Designed as a first stage rotor for a high pressure ratio compressor

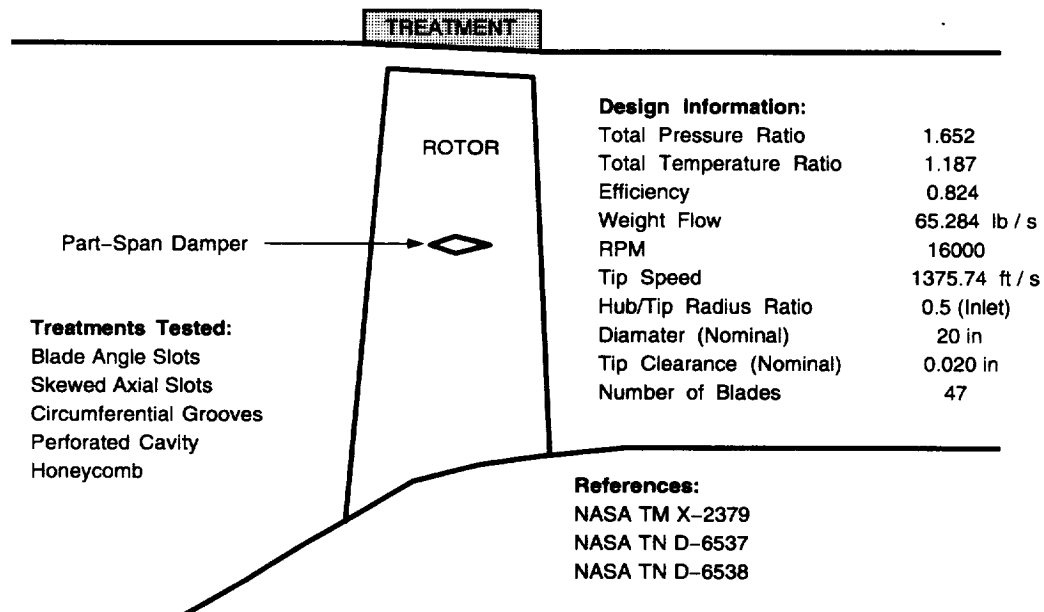


Figure 7.9: NASA Rotor 5 geometry and design performance summary.

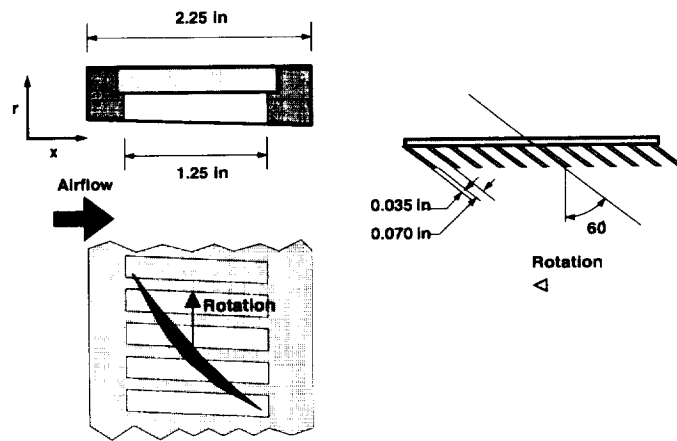


Figure 7.10: NASA Rotor 5 skewed axial slot endwall treatment geometry.

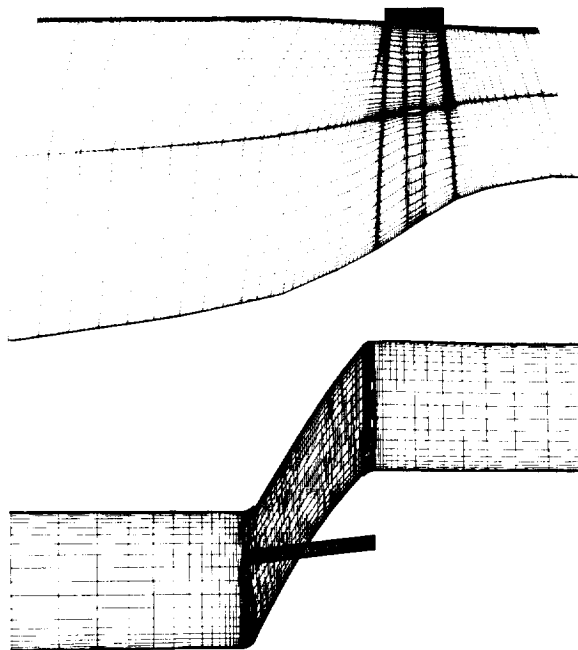


Figure 7.11: NASA Rotor 5 skewed axial slot endwall treatment geometry meridional and blade to blade mesh system.

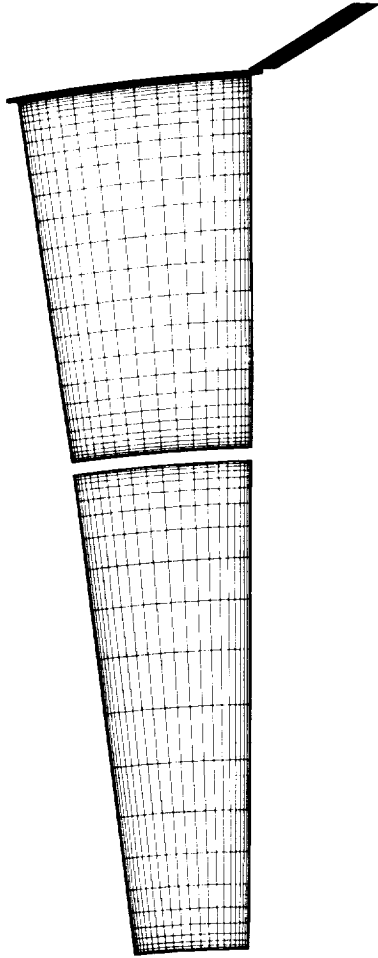


Figure 7.12: NASA Rotor 5 skewed axial slot endwall treatment geometry axial plane mesh system.

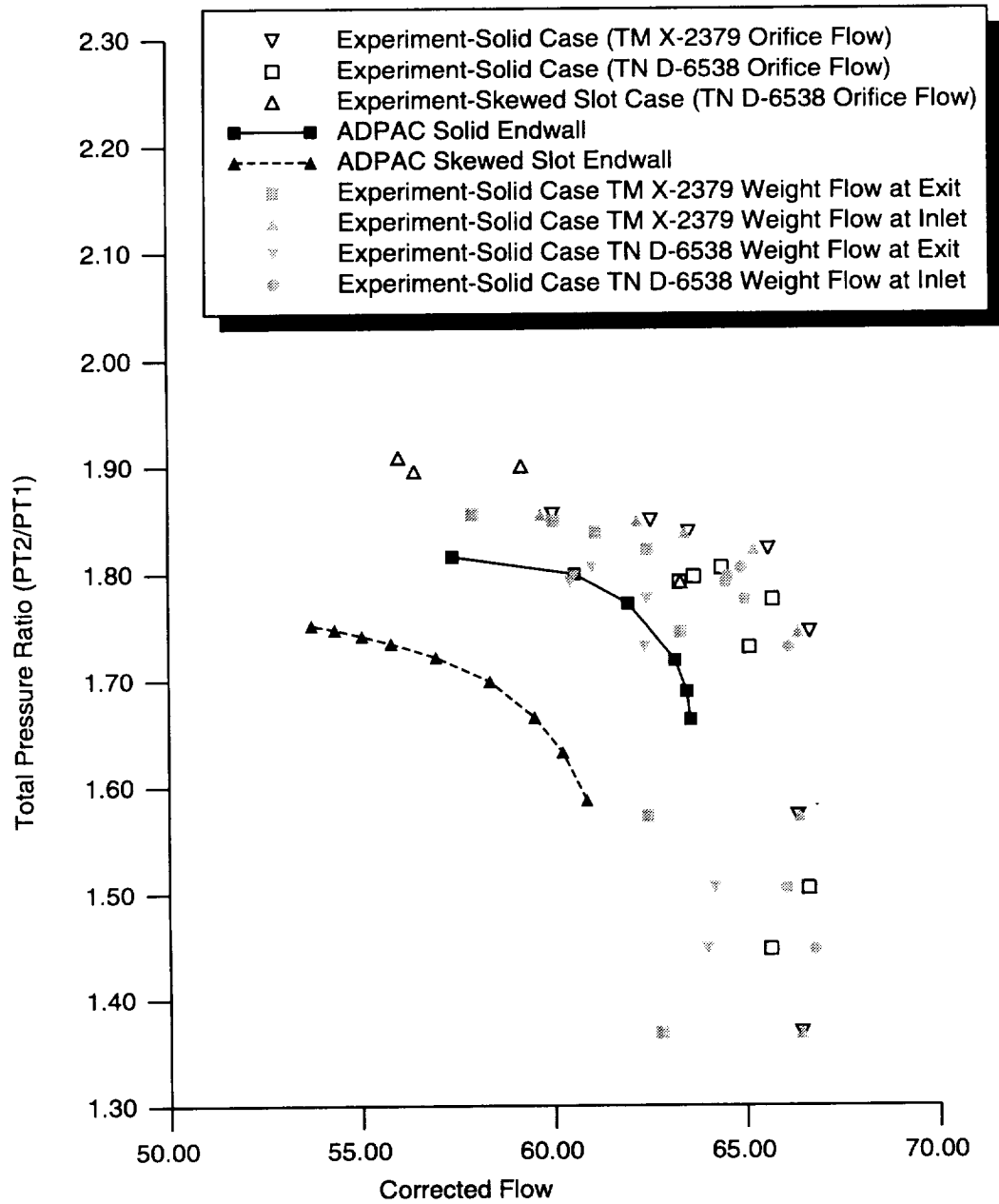


Figure 7.13: Comparison of predicted and experimental 100% speed pressure ratio versus mass flow operating characteristic for NASA Rotor 5 with and without skewed axial slot endwall treatment.

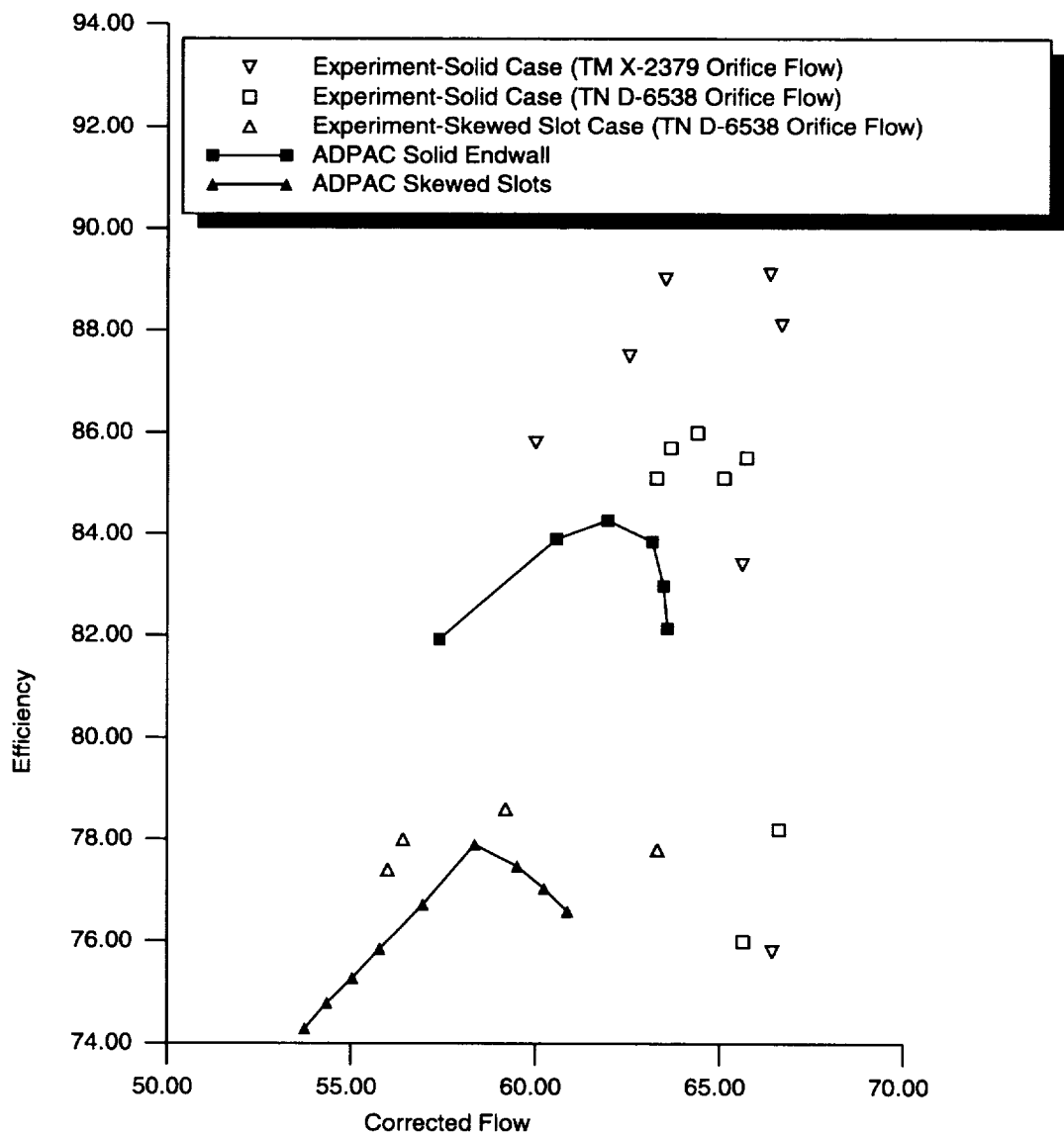


Figure 7.14: Comparison of predicted and experimental 100% speed efficiency versus mass flow operating characteristic for NASA Rotor 5 with and without skewed axial slot endwall treatment.

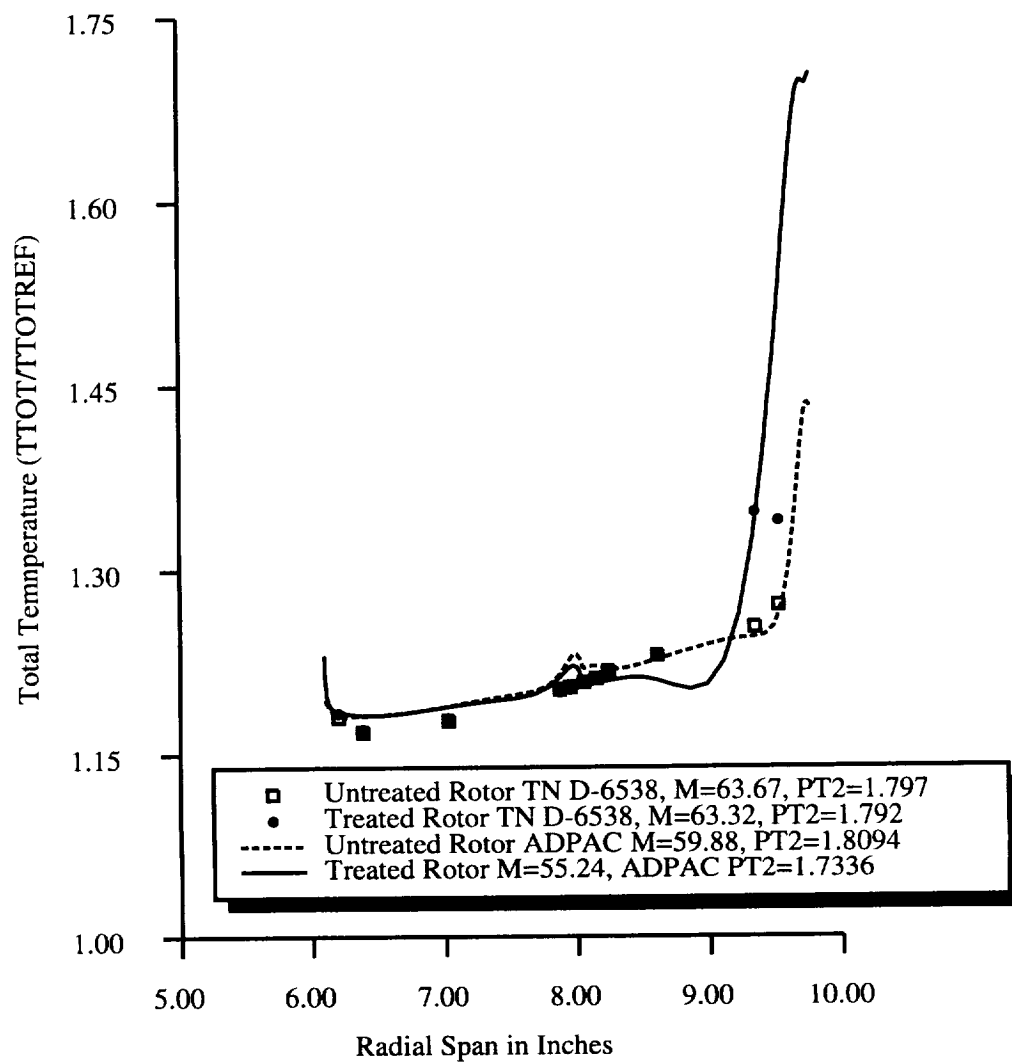


Figure 7.15: Comparison of predicted and experimental radial total temperature distributions for NASA Rotor 5 with and without skewed axial slot endwall treatment.

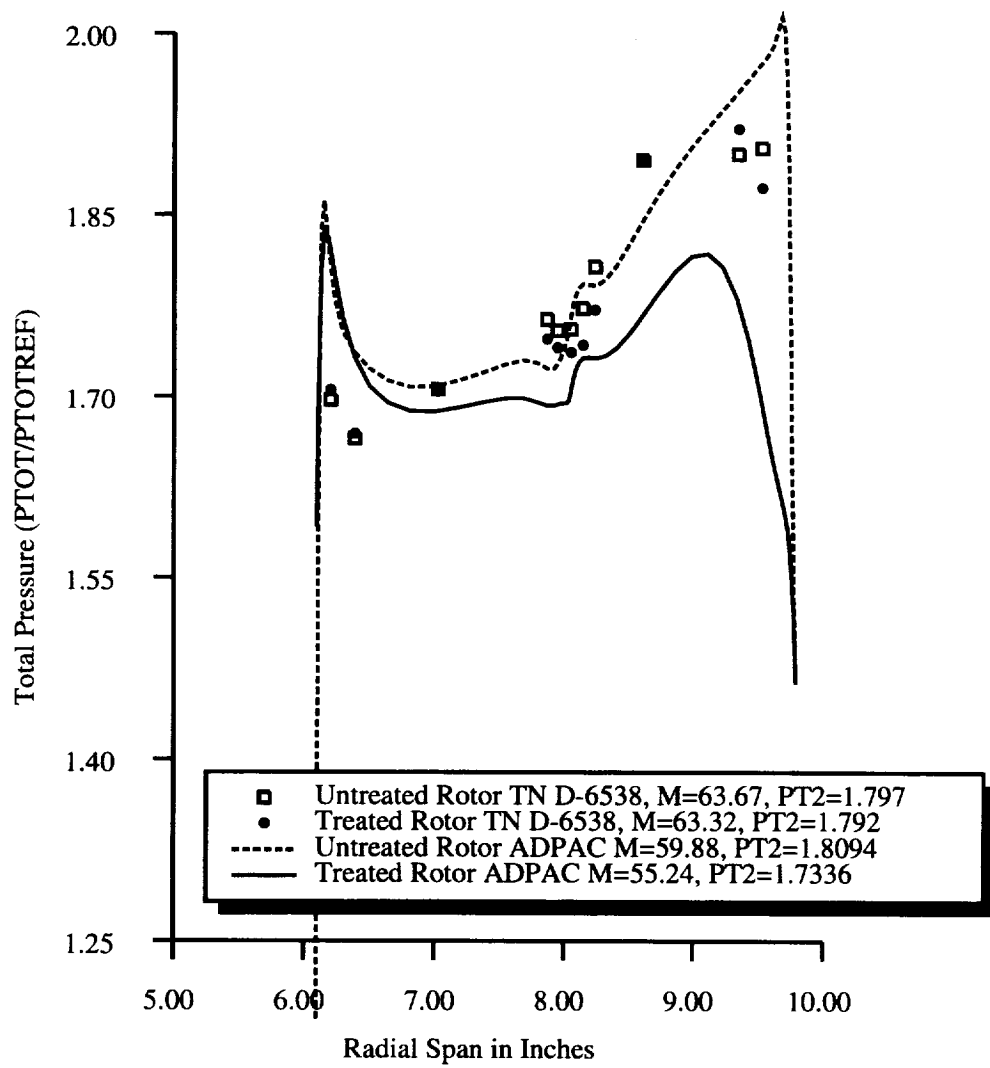


Figure 7.16: Comparison of predicted and experimental radial total pressure distributions for NASA Rotor 5 with and without skewed axial slot endwall treatment.

presence of the part span shroud. The tip clearance region was effectively represented by reducing the blade thickness to zero and employing a circumferential periodicity boundary condition radially outboard from the blade tip. Once again, the axisymmetric coordinates (x, r) of the upper blade passage mesh and the treatment mesh are coincident along the casing interface to facilitate the use of the endwall treatment time-average boundary condition. The mesh block sizes were 145x25x25 (axial, radial, circumferential for the inner blade passage mesh), 145x41x25 (outer blade passage mesh), and 105x25x17 (treatment slot mesh).

A comparison of the 100% design speed predicted and experimental total pressure ratio versus mass flow characteristic for NASA Rotor 5 is given in Figure 7.13 for both the baseline (no casing treatment) and endwall treated geometries. Before discussing the analysis, several anomalous data items must be noted. A survey of the various references describing experimental results for Rotor 5 was performed to check the consistency of the “baseline” (no casing treatment) flow measurements from one experimental study to the next. Mass flow rates were measured using several techniques during each experimental study. The data indicate that the investigators were unable to duplicate the mass flow characteristics from one build of the test rig to the next. Indications were [38] that it was impossible to reproduce the running rotor tip clearance between builds which contributed to the consistency problems. For a given build, mass flow rate measurements from the various techniques were also inconsistent, which leads to additional confusion. For completeness, several experimental data curves are included on Figure 7.13 to illustrate the variation in data from one run to the next. The computed Rotor 5 baseline speedline shows some qualitative agreement with the experimental data, but detailed comparisons were hampered by questions regarding the running blade geometry and tip clearance. Peak pressure rise and choke flow were both somewhat under predicted by the analysis in spite of the relatively fine meshes used. The computed rotor constant speed characteristic for the axial skewed slot casing treatment runs are also plotted on Figure 7.13. The predicted effect of the treatment was to reduce the mass flow rate, although a slight improvement in the rotor weight flow stall margin was observed. This improvement was, at least in relative terms, consistent with the experimentally observed behavior. Unfortunately, a corresponding reduction in pressure ratio was not consistent with the experimental trends, and this observation led to additional concerns for the endwall treatment time-average boundary condition. A similar comparison of predicted and experimental efficiency versus mass flow rate operating characteristics is given in Figure 7.14. One favorable observation was that the predicted relative changes in stall margin mass flow and adiabatic efficiencies resulting from the application of the casing treatment were accurately captured by the numerical simulations.

An analysis of the spanwise variation in flow properties for NASA Rotor 5 both with and without casing treatment is provided in Figures 7.15 and 7.16. In these curves, the predicted and experimental radial distributions of total temperature and total pressure versus span are compared for the baseline and endwall treated rotors near the baseline rotor stall point. It is interesting to note that total temperatures are relatively well predicted across the entire span for both the baseline and treated rotor geometries. The data for the treated geometry displayed a large increase in total temperature near the rotor tip which is indicative of the energy addition resulting from the treated endwall. Unfortunately, the comparison of spanwise total pressure distribution is not as favorable. The prediction for the endwall treatment geometry displays a rather large decrease in total pressure near the rotor tip which can only be attributed to the effects of the endwall treatment time-average boundary condition. The experimental data for this case indicate a slight *increase* in total

pressure near the tip. It seems apparent that the effect of the numerical endwall treatment time-average boundary condition provides the proper energy input to the blade passage flowfield, but that this energy is introduced in a relatively inefficient manner (presumably a result of the circumferential averaging inherent in this scheme) resulting in the noticeable degradation of total pressure near the tip.

7.3 AE3007 Fan Rotor Treatment Configuration Study

This section describes results from a treatment configuration study applied to a modern high bypass ratio fan rotor. The objective of this study was to analyze the effectiveness of the computational analysis in a realistic design-like scenario. Although extensive experimental data were not available, the results are nevertheless interesting and serve to demonstrate the applicability of complex 3-D aerodynamic analysis for conceptual design problems. The Allison AE3007 fan was selected as the baseline rotor for this study. The rotor selected is a relatively recent design based in large part on 3-D Navier-Stokes design procedures. The treatment configuration study examined the effects of circumferential groove, axial and blade angle slot, and recessed vane casing treatments on the performance of the AE3007 fan. The circumferential groove casing treatment was based on an Allison design for which limited experimental data were available. The treatment consisted of 5 circumferential grooves of moderate depth equally spaced from roughly 15% to 85% axial chord. The axial and blade-angle slot casing treatment geometries were designed to be consistent with primary geometric characteristics of the circumferential groove design (e.g. treatment depth), and followed many of the recommendations for casing treatment design (percentage of exposed area, axial breadth, etc.) given in Reference [39]. The recessed vane casing treatment was designed based on geometries used effectively for low speed axial fans [13].

Suitable meshes similar to those previously described were generated for each configuration, and a series of 100% constant speed operating curves were generated using the appropriate analysis. The circumferential groove casing treatment analysis utilized the direct couple boundary condition procedure (**PATCH**, as defined in the *ADPAC'07* User's Manual [28]), while the axial slot, blade-angle slot, and recessed vane treatment analyses were based on the endwall treatment time-average boundary condition (**ENDTTA**, as defined in the *ADPAC'07* User's Manual [28]).

Predicted 100% speed pressure ratio versus mass flow and efficiency versus mass flow characteristics are presented in Figure 7.17 for the circumferential groove casing treatment. The predicted and experimental baseline rotor performance is included on this plot for reference. A limited amount of experimental data was available for this particular configuration, and these data are also included. The predicted rotor baseline performance agrees fairly well with the measured performance curves. It is particularly satisfying that the stall-limit mass flow margin appears to be accurately captured. From the experimental data, the circumferential groove casing treatment was observed to extend the stall margin appreciably. By comparison, results from the analysis show a similar stall margin improvement, and, in fact, match the experimentally determined stall mass flow rate quite well. Results from a second series of analytical results based on the endwall treatment time-average boundary condition are also provided on Figure 7.17 for comparison. A circumferential groove represents the limit of a discrete treatment slot where the number of treatments equals the number of blades, and the treatment circumferential pitch is equal to the blade pitch. The

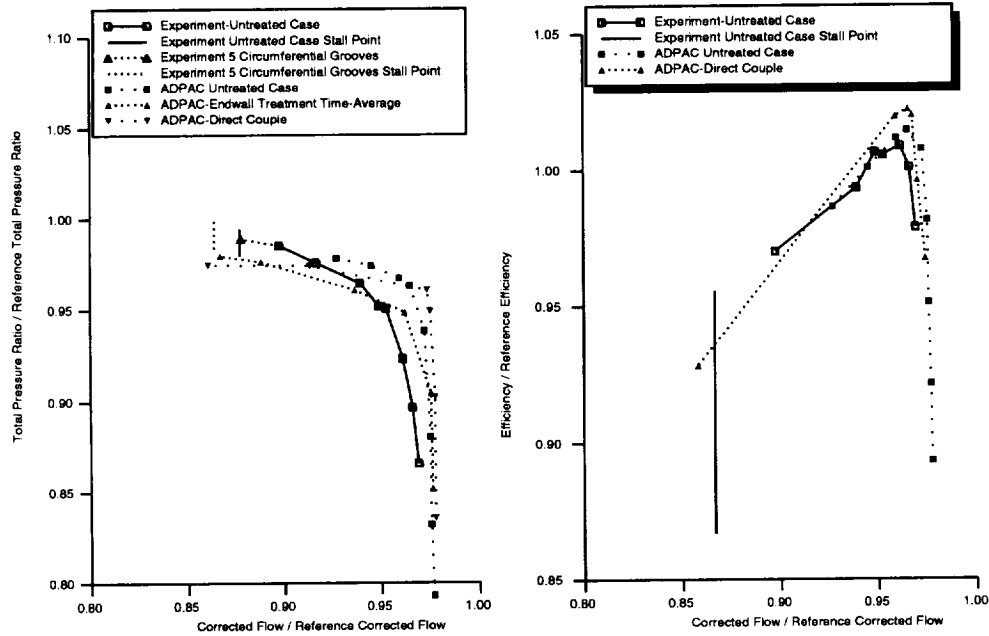


Figure 7.17: Comparison of predicted and experimental 100% speed total pressure ratio and efficiency versus mass flow operating characteristic for AE3007 fan with circumferential groove casing treatment.

endwall treatment boundary condition applied in this manner, and compared with the direct couple boundary condition results represents a validity check on the endwall treatment time-average boundary condition. Predictions from both boundary conditions were in good agreement, and resulted in nearly identical predicted stalling mass flow rates.

Similar performance curves are provided for the axial slot, blade-angle slot, and recessed vane casing treatments in Figures 7.18, 7.19, and 7.20, respectively. No experimental data were available for these configurations, so these results must be interpreted with the known limitations and deficiencies of the model in mind. The primary effect of the axial slot casing treatment was a reduction in pressure ratio near stall, with a larger stall margin improvement compared to the circumferential groove casing treatment. Efficiency was degraded over the entire operating curve. It should be noted that a slightly different rotor baseline performance was predicted using this mesh system compared to the circumferential groove mesh system. For each of the configurations in this study, the blade passage mesh changes significantly with treatment geometry, and the results appear to be sensitive to the mesh, particularly near stall. The difficulties associated with predicting stall have already been discussed, and these results serve to emphasize this point. A similar analysis of the blade angle slot casing treatment results indicate a less effective improvement in stall margin, but an unexpected increase in efficiency near stall. Previous experience with the endwall treatment time-average boundary condition has indicated that while the pressure ratio is not normally well predicted, the stall margin and efficiency often are, and these results are therefore presented with guarded confidence. Significant changes in the shock structure were observed for both the axial and blade angle treatment calculations compared to the baseline geometry. In most cases, the blade passage shock near the tip was constrained farther back in the blade passage due to the sharp corner of the treatment cavity. At certain pressure

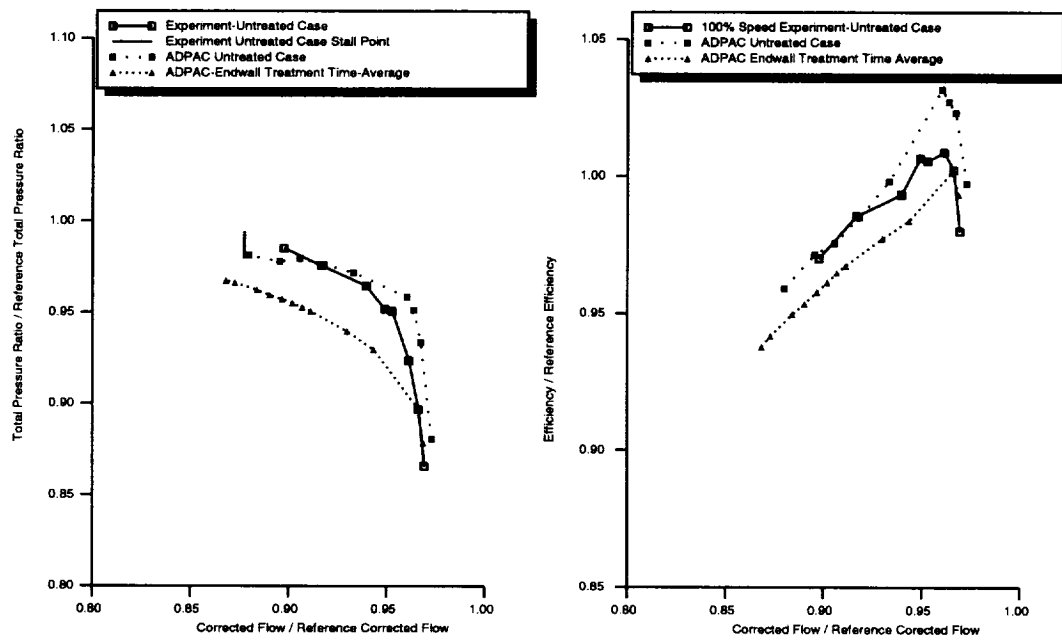


Figure 7.18: Comparison of predicted and experimental 100% speed total pressure ratio and efficiency versus mass flow operating characteristic for AE3007 fan with axial slot casing treatment.

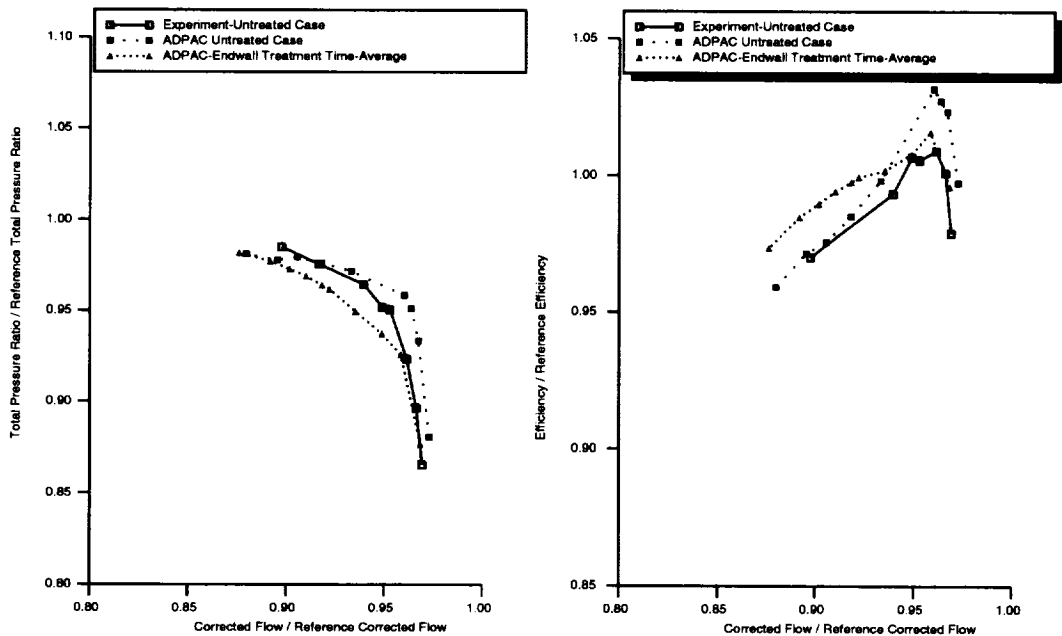


Figure 7.19: Comparison of predicted and experimental 100% speed total pressure ratio and efficiency versus mass flow operating characteristic for AE3007 fan with blade-angle slot casing treatment.

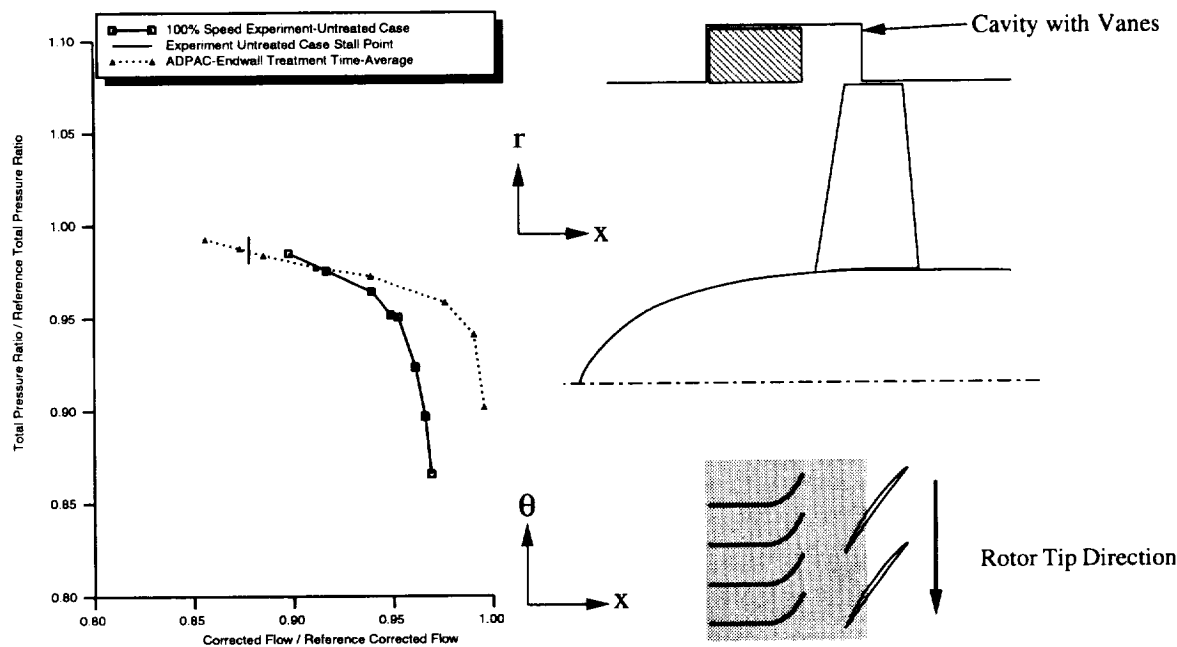


Figure 7.20: Comparison of predicted and experimental 100% speed total pressure ratio versus mass flow operating characteristic for AE3007 fan with recessed vane casing treatment.

ratios, a double shock appeared, and the relationship between treatment type and shock structure must not be overlooked as a mechanism for stall control of high speed fan rotors.

The recessed vane casing treatment results presented in Figure 7.20 illustrate a more dramatic change in performance compared to the baseline rotor results. The drastic modification of the endwall and the open area over the rotor effectively uncovers the throat region, resulting in a significant change in rotor choked flow. The predictions indicate a rather large stall flow margin, but the unconventional nature of this geometry suggests that further study is required.

A comparison of the predicted blade surface pressure distributions and clearance flow particle traces is given in Figures 7.21-7.25 for each of the 5 geometries (solid endwall, circumferential groove, axial slot, blade angle slot, and recessed vane), respectively. From these figures it is apparent that there are drastic differences between treatment types and the resultant effects on blade surface pressure and clear flow trajectories. The primary effect of the axial slot, blade-angle slot, and recessed vane treatment predictions appears to be a sweeping of the clearance flow downstream, thus eliminating much of the blockage associated with the clearance flow in the forward part of the blade passage. This results in a significant alteration of the near tip shock structure for this transonic rotor. The axial slot, blade-angle slot, and recessed vane treatment predictions clearly show a double passage shock resulting from the strong flow recirculation due to the treatments, as well as the sharp corners in the flow path associated with the treatment cavities. The circumferential groove predictions show some of the same clearance flow behavior, but to a lesser extent, and the blade passage shock appears to have moved slightly downstream near the tip compared to the baseline (solid endwall) prediction. It is unclear from these solutions whether the primary stall margin enhancement afforded by the treatment is a result of the clearance flow manipulation or the change in shock structure and location.

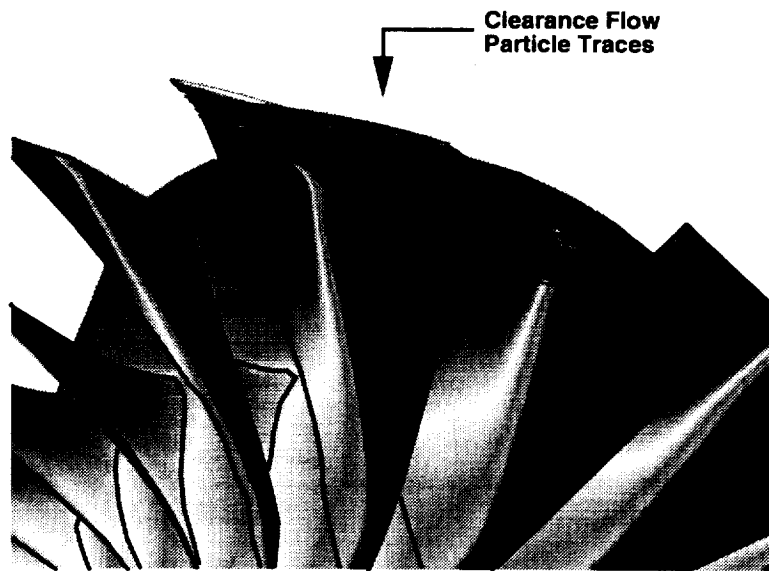


Figure 7.21: Predicted clearance flow particle traces and rotor surface static pressure contours for AE3007 fan with solid endwall (no treatment).

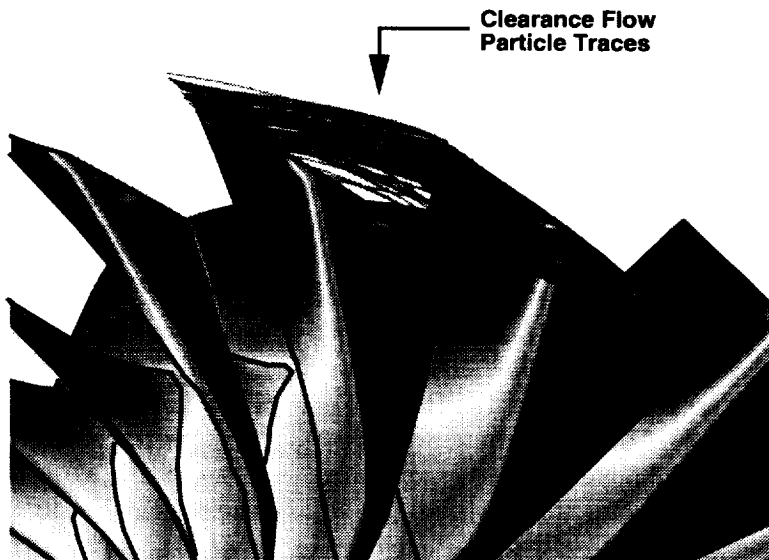


Figure 7.22: Predicted clearance flow particle traces and rotor surface static pressure contours for AE3007 fan with circumferential groove endwall treatment.

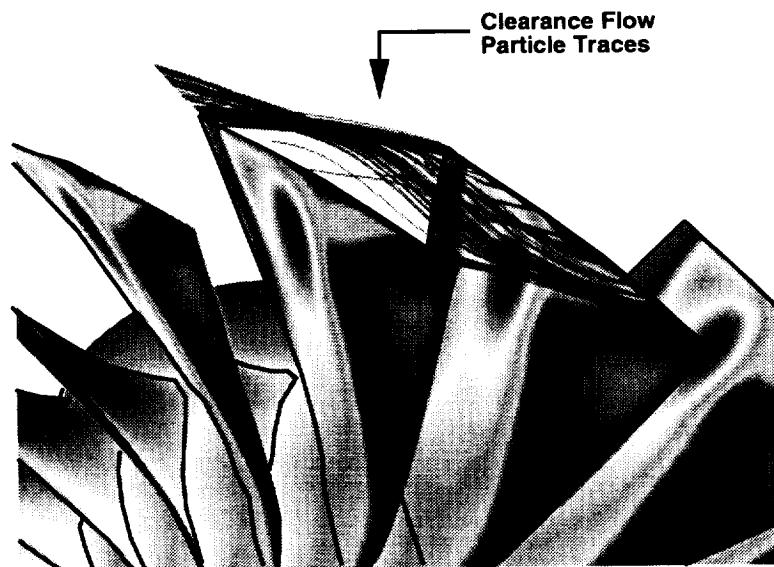


Figure 7.23: Predicted clearance flow particle traces and rotor surface static pressure contours for AE3007 fan with axial slot endwall treatment.

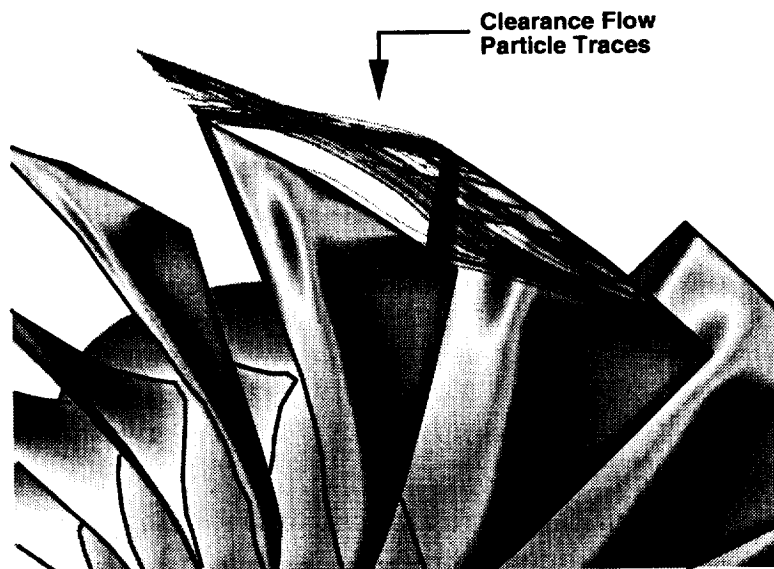


Figure 7.24: Predicted clearance flow particle traces and rotor surface static pressure contours for AE3007 fan with blade-angle slot endwall treatment.

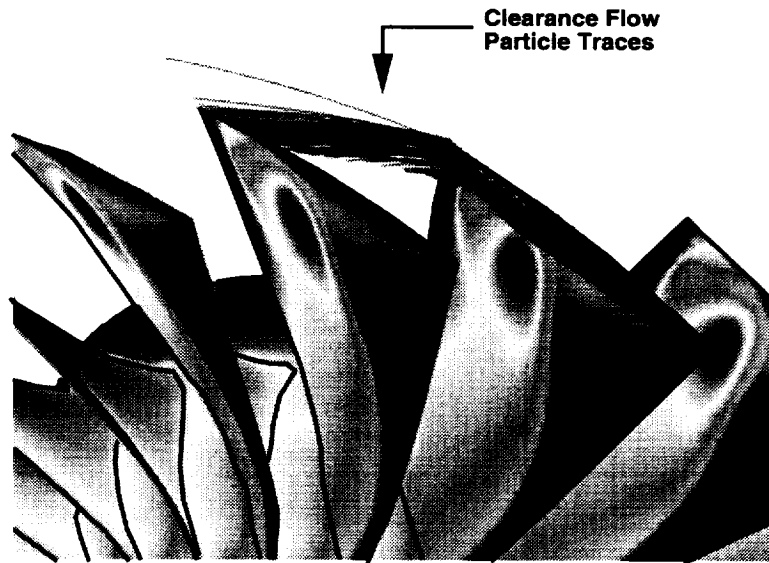


Figure 7.25: Predicted clearance flow particle traces and rotor surface static pressure contours for AE3007 fan with recessed vane endwall treatment.

During the course of the treatment configuration study, it was determined that a simple effective tool for examining the effect of the treatment on the overall blade passage flow was to circumferentially average the final predicted flowfield and examine the predicted axisymmetric flow velocity vectors near the tip of the rotor. Several plots of this type are given in Figure 7.26 for the baseline and treated rotor flowfields for the baseline rotor near stall operating point. In the case of the baseline rotor, the axisymmetric average of the clearance flow vortex is manifested as a recirculation bubble in the clearance region. Numerical experimentation has suggested that as this representative recirculation bubble moves farther and farther upstream, the rotor approaches the stall point. The remaining plots illustrate a similar analysis of the AE3007 fan rotor with the selected endwall treatment applied. It is readily obvious that for each configuration, the endwall and clearance flows are rather drastically altered. In the case of the 5 circumferential groove casing treatment, the large recirculation bubble observed in the baseline case is now represented as 5 smaller recirculation bubbles confined to the vicinity of the grooves. It is speculated that this confinement is one factor contributing to the stall margin improvement observed with circumferential groove casing treatments. The endwall flow behavior for the remaining treatments are even more dramatic, displaying large confined regions of recirculating flow which contribute to the stall margin enhancement process.

One important observation from this study was that the near stall flowfield for both the baseline and each of the treated rotor geometries displayed similar near leading edge endwall flow characteristics. The results of this study supported the observations on the role of clearance flows on stall behavior reported in [37]. The present predictions suggest that the mechanisms of stall were much the same for both the baseline and treated configurations, but that stall occurred at different mass flow rates. These results indicate, at least conceptually, the usefulness of a detailed casing treatment predictive tool for design analysis.

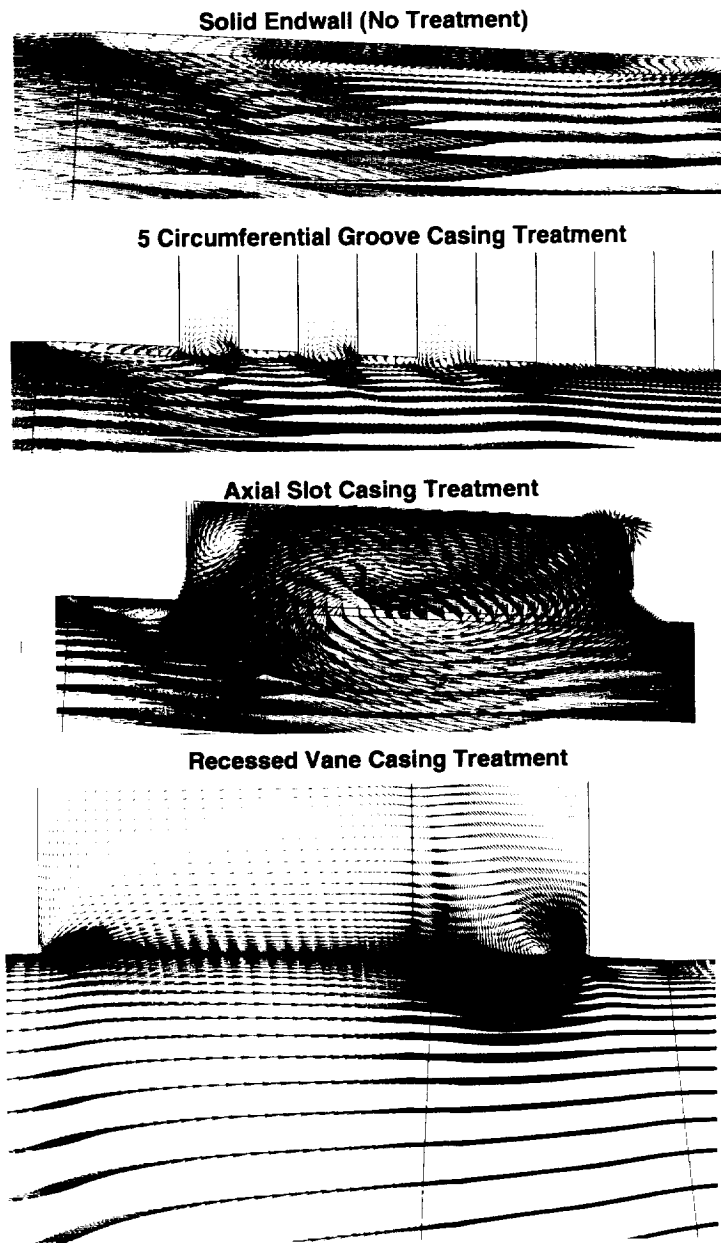


Figure 7.26: AE3007 fan treatment configuration study summary of axisymmetric averaged endwall flow behavior.

7.4 Summary of Steady State No Distortion Predictions

Detailed steady state aerodynamic predictions have been obtained for various compressor endwall treatment configurations using an advanced 3-D Navier-Stokes analysis technique. Numerical boundary conditions developed for the steady state analysis include a direct couple scheme for circumferential groove endwall treatments and an endwall treatment time-average approach for predicting steady state flows for discrete slotted endwall treatments.

The analysis of the MIT stator demonstrated the stability of the endwall treatment time-average boundary condition, but the numerical results indicated that for this low speed flow, the beneficial effects of the treatment were somewhat under predicted by the analysis. This discrepancy was believed to be due (at least in part) to the circumferential averaging procedure used in the numerical approximation.

The analysis was subsequently applied to a high speed flow case utilizing the NASA Rotor 5 geometry with skewed axial slot casing treatment. A detailed analysis of these results indicated that the numerical representation of the discrete treatments provided an adequate increase in total temperature (representative of energy input), but not in total pressure (representative of efficiency).

A detailed design configuration study was performed for the AE3007 fan rotor with circumferential groove, axial slot, blade angle slot, and embedded vane endwall treatments. The results of the study indicated that circumferential grooves should be used as a conservative approach (minimal loss in efficiency) to enhance stall margin, and that additional stall margin improvement would most effectively be afforded by axial slots.

The discrepancy between the endwall treatment time-average predictions and the available experimental data imply that additional developments are required for this type of boundary condition before accurate design analyses can be performed. It is also possible that some of the beneficial effects of the discrete slot endwall treatments result from a truly time-dependent flow structure, in which case only the time-accurate solution scheme can be expected to accurately capture this flow. Fortunately, these problems do not detract from the validity of the solutions involving circumferential groove treatments.

Chapter 8

TIME DEPENDENT CASING TREATMENT ANALYSIS (NO DISTORTION)

Several preliminary 3-D calculations were performed for the AE3007 Fan rotor with a recessed vane casing treatment using the time-accurate boundary condition procedure described in Chapter 3. The results in this section were derived primarily under NASA Contract NAS3-25270, Task 6 - Casing Treatment Analysis, but are reported formally under the present task.

8.1 NASA Rotor 5 with Axial Skewed Slot Casing Treatment

The first test case is based on the calculation of the time-dependent aerodynamic interaction resulting from the relative motion between a rotating fan blade and a discrete endwall treatment geometry. This test case was based on the NASA Rotor 5 fan with axially skewed slot endwall treatments. The NASA Rotor 5 design and endwall treatment geometry are described extensively in the previous chapter. The ratio of endwall slots to rotor airfoils was 6:1, and therefore a single blade passage representation is used in conjunctions with 6 endwall treatment slot representations. In order to permit communication between the endwall treatment mesh systems and the rotor blade passage mesh, an intermediate H-type mesh was constructed as shown in Figure 8.1. The intermediate mesh shares common points with the inner portion of each of the treatment meshes. The intermediate mesh also shares common axial and radial points with the outer portion of the blade passage mesh in the tip clearance flow region. The interface between the blade passage mesh and the intermediate mesh is therefore treated as a sliding boundary, and flow data is transferred from the blade passage mesh to the intermediate mesh (and subsequently the treatment slots) through a time-space interpolation procedure. The interface between the blade passage mesh and the intermediate mesh is equivalent to the inter-blade row interface in a rotor/stator aerodynamic interaction analysis for multistage turbomachinery.

The solution was initiated from a steady state analysis for the equivalent geometry with no relative motion between the rotor and treatment. At that point, the time-accurate solution was advanced until a time-periodic solution was obtained. Figure 8.2 illustrates an

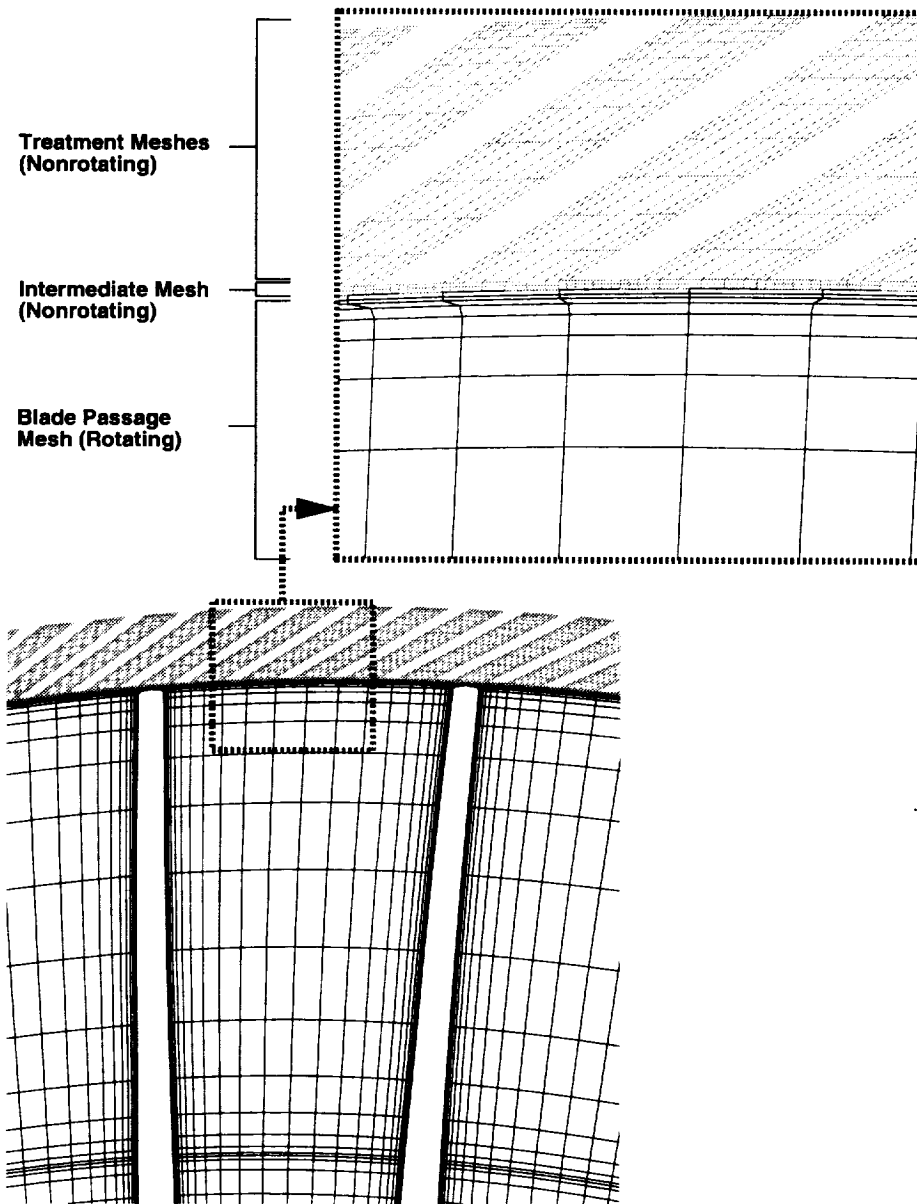


Figure 8.1: NASA Rotor 5 mesh summary illustrating intermediate mesh used to couple blade passage and treatment passage mesh systems.

instantaneous snapshot of the rotor/treatment interaction solution. Static pressure contours on an axial plane are highlighted to show the interaction between the rotor passage flow and the treatment slots. Static pressure contour lines are also given on a radial plane near the tip to illustrate the large gradients associated with the leading edge stagnation point and bow shock flow features. A significant amount of flow injection/removal was observed in the numerical solution for the rotor/treatment aerodynamic interaction. The flow removal occurred primarily over the aft 40% of the treatment slot. Flow injection occurred over the forward 35% of the treatment slot, which suggests that approximately 25% of the open treatment slot region was unused. The time evolution of the treatment slot flowfield is illustrated by the instantaneous velocity vector patterns given in Figure 8.3. The images presented on Figure 8.3 represent five “snapshots” of the time-periodic flow cycle experienced by each treatment slot. The blade passage flow is from bottom to top in this figure. The velocity vectors are shaded by absolute total pressure (dark arrows indicate low total pressure). The vectors represent the instantaneous velocity at points on the mid-pitch plane (circumferentially) of the treatment cavity. The flow in the slots essentially provides a recirculation path for the downstream, high total pressure flow to the forward portion of the rotor. Flow removal (from the blade passage perspective) occurs over the rearward 40% of the slot opening. This flow removal does not vary significantly over time. Flow injection occurs over the forward 40% of the treatment slot opening. The flow injection process is highly unsteady. As the rotor tip approaches the slot opening, the high static pressure on the blade retards the injection process from the slot. At some locations, this retardation is strong enough to result in very local flow removal from the blade passage. As the rotor passes the treatment opening, a large, rapid decrease in the pressure seen by the treatment opening occurs. A strong, jet-like flow injection occurs immediately after the rotor suction surface has passed the treatment slot opening. This flow injection has low axial velocity, but relatively large circumferential and radial velocities. The relative total pressure of the injected fluid is high because of the turning imparted by the skewed slot. The skewing of the slot essentially acts to turn the flow passing through the treatment cavity, much like a vane. The predicted amount of turning on a mass-averaged basis was approximately 43 degrees. The effect of this turning is illustrated in the comparison of the near blade tip loading interpreted from the blade surface static pressure ratio distributions for the baseline rotor (steady state, no treatment analysis) and the the time averaged loading for the rotor/treatment aerodynamic interaction analysis shown in Figure 8.4. The incidence change effected by the flow injection at the front of the passage serves to increase the rotor loading near the tip, and presumably the flow removal downstream aids to prevent flow separation due to the increased loading. Corner vortices are evident in all of the predicted velocity vector patterns, which suggest that the treatment passages should not be designed as rectangular slots.

8.2 AE3007 Fan Rotor with Recessed Vane Casing Treatment

A time-dependent analysis was also performed for the AE3007 fan with recessed vane casing treatment to examine the unsteady aerodynamics associated with the rotor/treatment passage interaction, and to provide a comparison between solutions at a common operating point using the endwall treatment time-average and time-dependent boundary conditions. The treatment geometry was purposely designed to have a 2:1 treatment vane to rotor

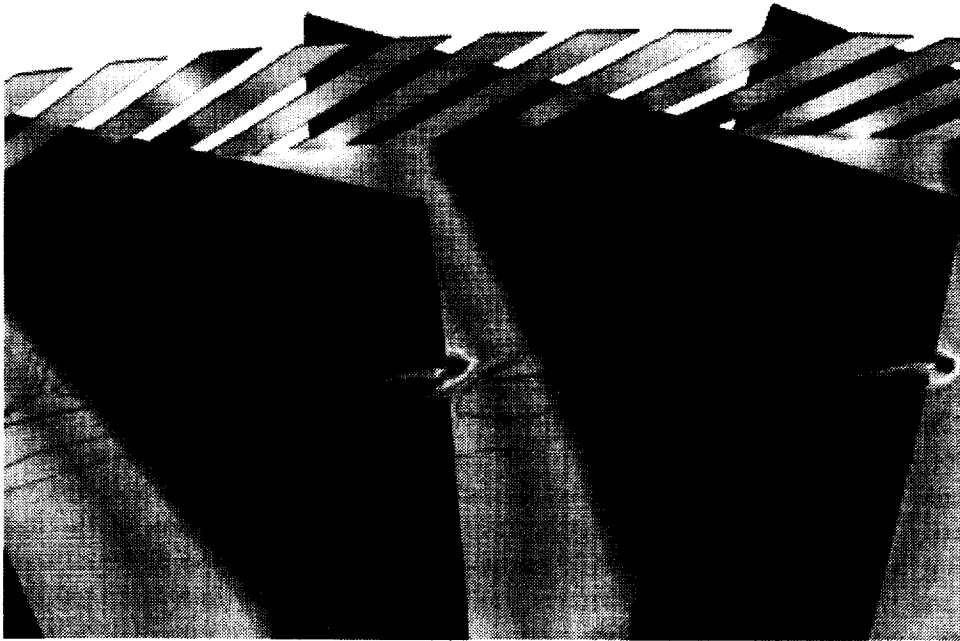


Figure 8.2: Predicted instantaneous static pressure contours (axial plane) and static pressure contour lines (radial plane) for NASA Rotor 5 rotor/endwall treatment aerodynamic interaction analysis.

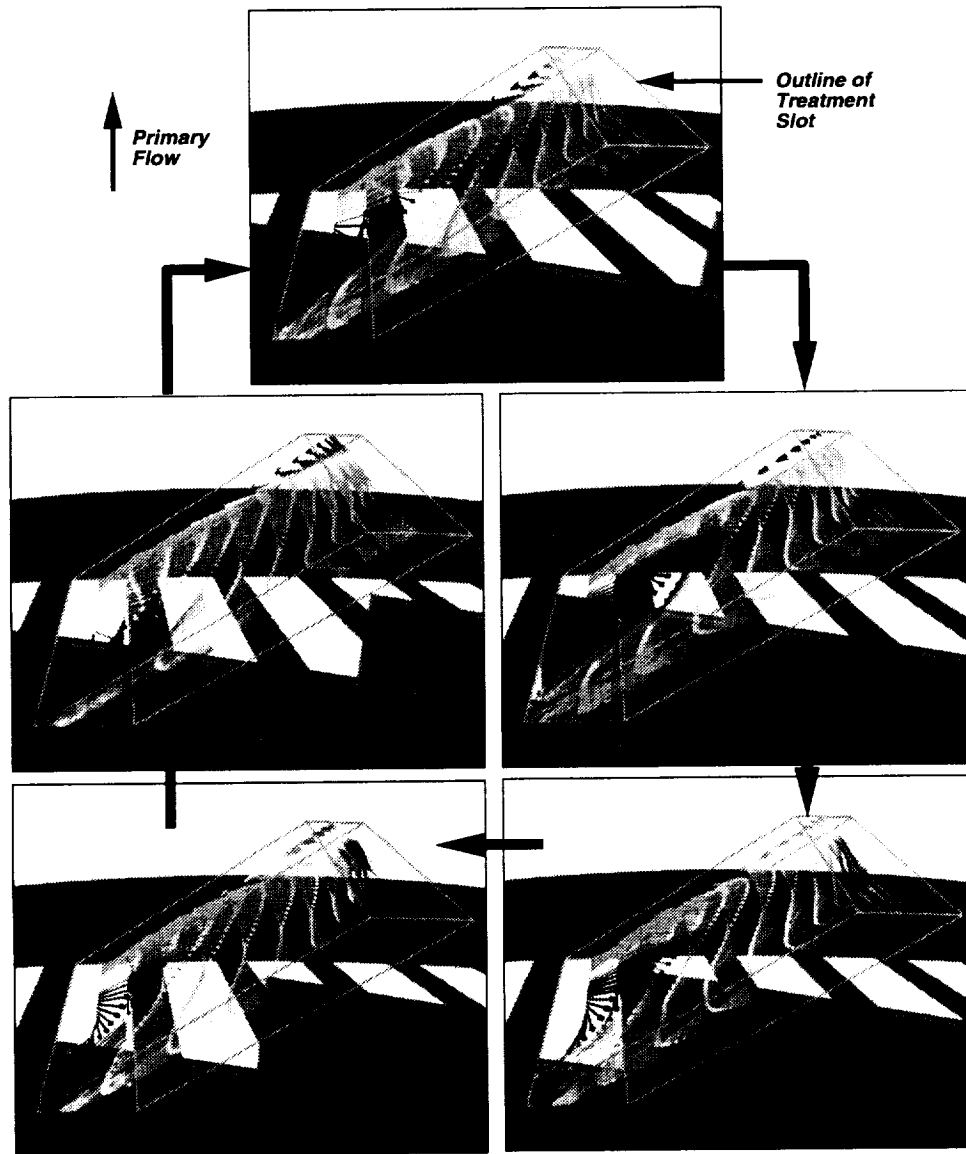


Figure 8.3: Predicted instantaneous treatment slot velocity vector patterns for NASA Rotor 5 rotor/endwall treatment aerodynamic interaction analysis.

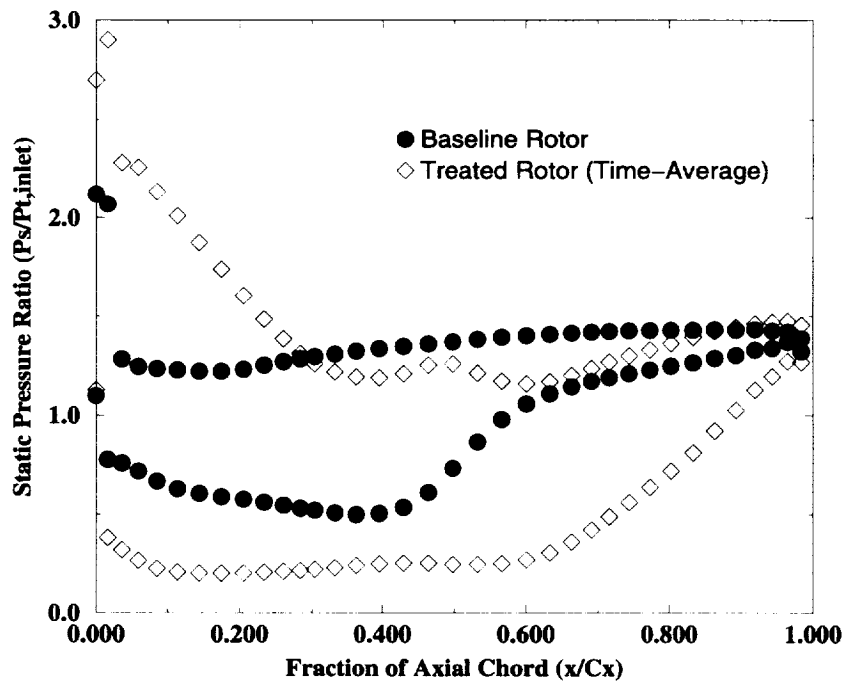


Figure 8.4: Comparison of predicted near rotor tip blade loading for baseline (no treatment) and treated (time-average of rotor/treatment interaction) solutions.

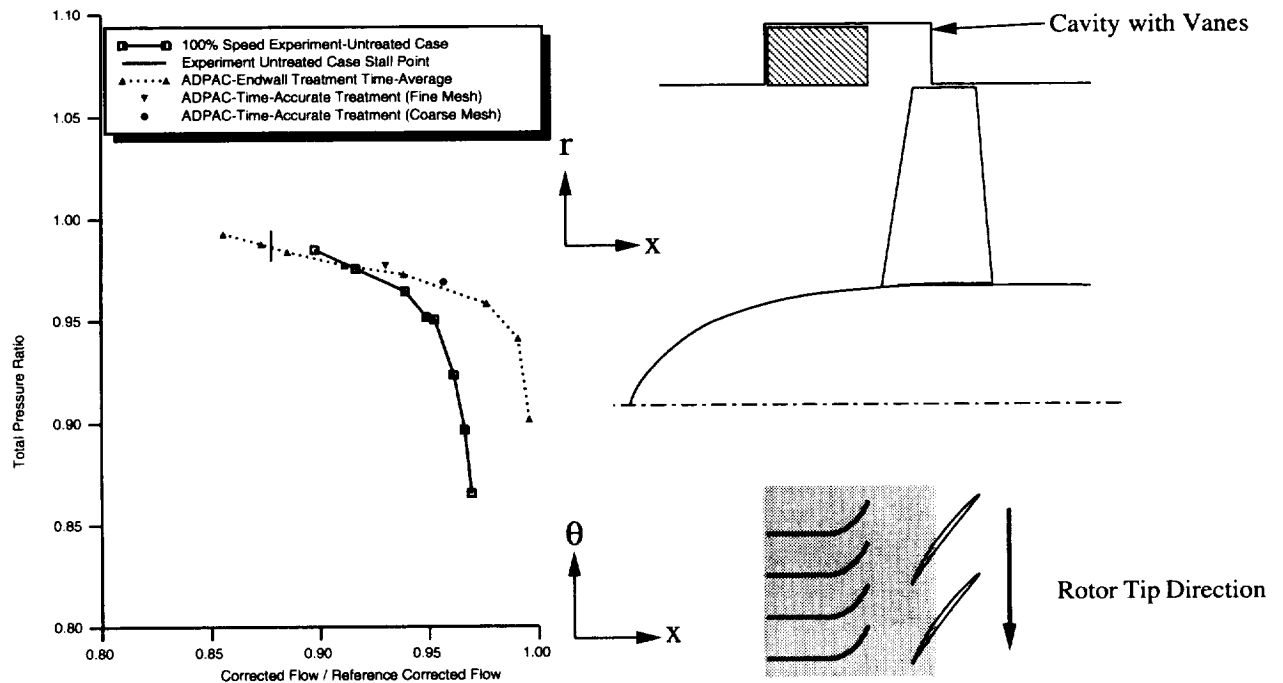


Figure 8.5: Comparison of time-dependent predicted and experimental 100% speed total pressure ratio versus mass flow operating characteristic for AE3007 fan with recessed vane casing treatment.

blade count ratio to simplify the mesh requirements for the time-dependent solution. This ratio permits the reduction of the problem to a single blade passage and 2 treatment vane passages. The analysis was applied to an operating point which was believed to be near the untreated rotor stall point. The time-marching solution was advanced through approximately 10 blade passage periods before the solution was deemed time-periodic.

Time-dependent predictions were performed for two mesh systems, the coarser mesh being obtained from the original mesh by removing every other mesh line in each coordinate direction. The time-dependent solutions were ultimately time-averaged, resulting in the operating condition indicated on the speed line in Figure 8.5. Both the fine mesh and the coarse mesh predictions demonstrated good agreement with the speed line obtained using the endwall treatment time-average boundary condition. An analysis of the unsteady flowfield indicated a time-periodic vortical structure emanating from the leading edge of the rotor, no doubt a result of the fact that the leading edge at the tip extends under the open cavity. The vortical structure behaves similar to a vortex shedding phenomena off a bluff body, except that the flow does not appear to be separated (in the traditional two-dimensional boundary layer sense).

Predicted static pressure contours on an axial plane within the blade passage are given in Figure 8.6. A portion of the blade case has been removed from this figure to expose the recessed vanes. The vortical structure previously mentioned is indicated by the alternating regions of high and low pressure above the rotor in the treatment cavity. The overall complexity of the flow cannot be explained in simple turbomachinery airfoil flow parameters.

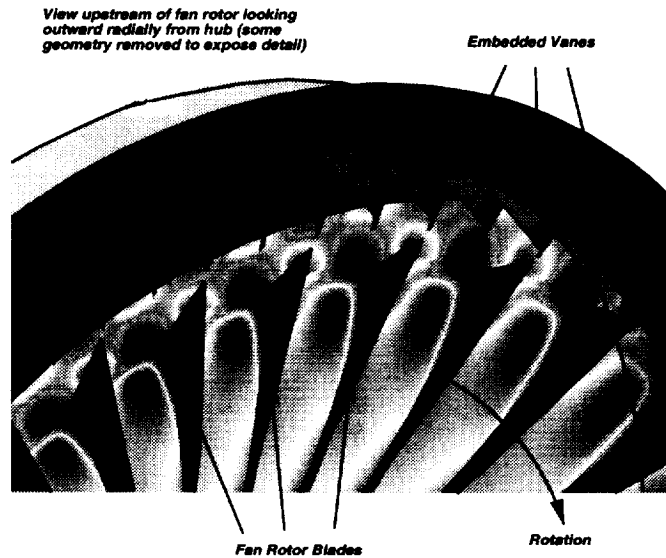


Figure 8.6: Predicted instantaneous static pressure contours for AE3007 fan rotor with recessed vane casing treatment (45% axial chord).

8.3 Summary of Time Dependent No Distortion Predictions

Detailed time-dependent aerodynamic predictions have been obtained for a compressor end-wall treatment configuration using an advanced 3-D Navier-Stokes analysis technique based on a time-accurate boundary interface procedure for predicting time-dependent airfoil/endwall treatment aerodynamic interactions.

A time-dependent solution of the aerodynamic interaction between the AE3007 fan rotor and an embedded vane casing treatment with a 1:2 rotor blade to treatment vane airfoil ratio was performed at the untreated rotor near stall condition. Time-averaged results from the unsteady analysis showed favorable agreement with steady state results using the endwall treatment time-average boundary condition. This suggests that poor predictions previously obtained with the endwall treatment time-average boundary condition were likely due to the fact that the treatments involved intermittent regions of open flow (treatment opening) and endwall. The present treatment application (recessed vane) does not involve intermittent endwall regions which might ultimately sway the circumferential averages. This apparent drawback of the endwall treatment time-average boundary procedure requires additional development.

The time-dependent analysis indicated a time-dependent periodic vortical flow structure near the rotor tip, although this was a rather extreme geometric departure from the original design. This unusual flow structure was apparently not a significant feature of the treatment aerodynamic effect, as comparisons between the time-average of the unsteady flow calculation and the steady state calculations based on the endwall treatment time-average

boundary condition displayed good agreement. The recess vane treatment demonstrated significant promise for extending the aerodynamically stable operating range of a high speed fan, and is apparently suitable for modeling using steady state solution techniques. These features indicate that future design efforts based on this type of treatment would benefit significantly from CFD analysis.

Chapter 9

STEADY STATE CASING TREATMENT ANALYSIS WITH INLET DISTORTION

In this section, several steady state predictions of turbomachinery blade row flowfields both with and without casing treatment are presented. The results presented in this section are intended to address the effects of inlet flow distortion. The results in this section were derived under NASA Contract NAS3-25270, Task 7 - Inlet Distortion Analysis.

9.1 Radial Flow Distortion Analysis

A series of calculations was performed to evaluate the performance of a modern turbofan engine fan rotor in the presence of inlet distortion both with and without compressor casing treatment. The geometry selected for this study was the Allison AE3007 Type III fan rotor with circumferential groove casing treatment. The AE3007 fan is a 38 inch diameter fan powering a 7,000 pounds thrust turbofan engine being developed for the regional airliner market. Previous calculations for a similar geometry were successful, and some experimental data (with distortion, but for an untreated rotor) were available at Allison to verify the predicted results. A 5 groove casing treatment geometry was defined and is illustrated in Figure 9.1. Mesh generation for this task was performed using the *TIGG3D* mesh generation program due to the simplicity with which the circumferential groove casing treatments may be applied (the circumferential grooves are represented as zero thickness, zero camber blade rows in the *TIGG3D* input). Mesh generation is considered a crucial factor in the success of this analysis due to the unique requirements for predicting details of a radially distorted flow. The radial flow distortions were represented by a degradation in inlet total pressure concentrated near the tip of the fan. Previous experience has shown that the ability to accurately convect a radially distorted flow requires a large concentration of grid points in the vicinity of the distortion. Therefore, the meshes used in this study have been generated with a tip-heavy concentration of mesh points.

9.1.1 Introduction

Within this section, the predicted effects of tip radial distortion inflow will be presented as they relate to the operation of a modern turbofan engine fan rotor, and what effects, if any,

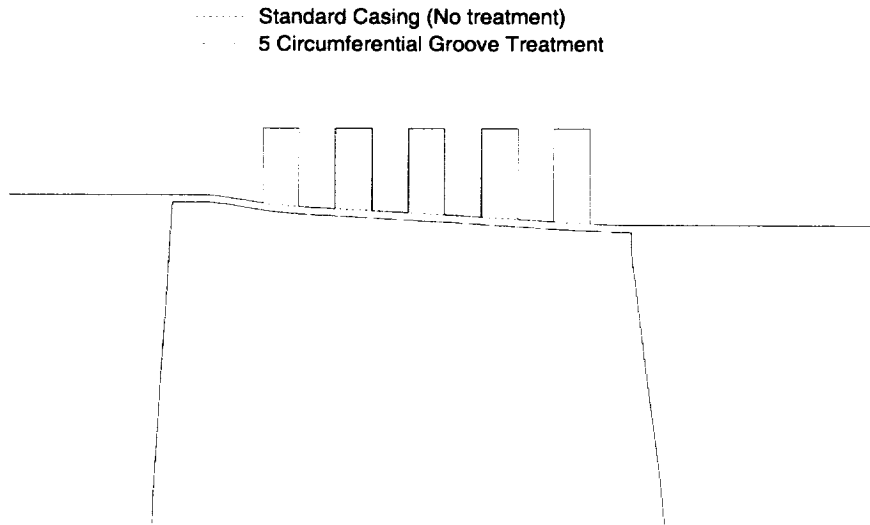


Figure 9.1: Comparison of the standard (no groove) casing with the 5 groove endwall casing treatment geometries.

an endwall casing treatment will provide. The flow prediction code *ADPAC07* was used to obtain solutions for the Allison AE3007 fan rotor for cases with and without distorted inflow. In combination with the variation in the inflow profiles, the effects of an endwall casing treatment were also studied. The endwall treatment chosen was comprised of five circumferential grooves above the tip of the blade as shown in Figure 9.1. The variables in the geometry and inflow profile led to four separate test cases: flow with and without casing treatment, and flow with and without tip radial inflow distortion.

The following sections review the grid generation process, the inlet profiles used, the solution procedure, the results obtained from *ADPAC07* including a comparison with experimental data, and the effects of varying grid density on the solution.

9.1.2 Grid Generation

The following mesh requirements were defined prior to evaluating mesh size and structure. The grid must provide adequate resolution to define the blade geometry and the casing endwall treatment. Due to the testing of tip radial distortion cases, the grid spacing should be tighter near the tip of the blade and through the tip clearance region. Another factor considered in the grid selection was that calculations both with and without casing treatment should be performed on the same blade passage mesh. Finally, it was required that the blade passage mesh and the groove mesh systems have contiguous mesh distributions along their common boundary.

The grid generation code *TIGG3D* was used to create an H-type grid which satisfied the desired criteria described above. The grid was initially generated as one large block including the five casing treatment grooves. This large grid was then split into a main

blade passage grid and five separate groove grids. In order to provide for a sharp edge on the tip of the blade and to avoid the grid shear normally associated with blade tips for an H-type mesh, the blade passage airfoil mesh surface was generated as if no clearance region existed (the blade surfaces extend all the way to the case). A separate tip clearance grid was then generated to match the blade passage in the clearance region. This structure eliminated the rounding of the blade edges. The casing treatment groove grids were then augmented to extend over both the blade passage and clearance region meshes. The final grid block counts for each case were two for the baseline rotor (no casing treatment) and seven for the five groove casing treatment configuration.

The maximum grid limits were selected to permit three levels of multi-grid during the *ADPAC07* solution. The final grid dimensions were 205 x 57 x 49 for the blade passage mesh, 157 x 9 x 13 for the tip clearance mesh, and 13 x 13 x 61 for each circumferential groove mesh. The blade surface point distribution was 157 points from leading edge (located at $i = 25$) to trailing edge and 49 points from hub to tip. The grid inlet was located at 1.8 chords upstream from the leading edge and the grid exit was located at 1.4 chords downstream of the trailing edge. The resulting numbers of grid points for each geometry were: 590,934 for the standard (no groove) geometry and 642,479 for the five groove geometry. An axisymmetric projection of the H-type mesh for the teated rotor configuration is illustrated in Figure 9.2. The mesh construction was algebraic, and utilizes hyperbolic functions to establish the mesh clustering. The maximum mesh cell expansion ratio for this mesh was 2.56 (in the clearance mesh).

The grid described by the dimensions above will be referred to as the *fine* grid. Solutions were also obtained on a less refined grid. This *coarse* grid was constructed by eliminating every other mesh line in each of the three coordinate directions from the fine mesh. This coarse grid was used to establish initial results and to provide a comparison of grid density sensitivity which will be addressed later within this report.

9.1.3 Distorted Inlet Profile

Solutions for each geometry were performed for two different inlet profiles referred to as the clean inlet and the tip radial distortion inlet. The clean inlet employed a uniform inflow profile with constant total temperature and constant total pressure equal to their respective reference values. The tip radial distortion inlet profile had a uniform radial total temperature distribution equal to the reference value, but with a radial total pressure distribution which was degraded near the tip. The total pressure began to decrease at approximately 70% span and decreased to a value of 87% of the reference total pressure. A comparison of the radial total pressure distributions for the two inlet profiles is shown in Figure 9.3. The distorted inlet profiles were implemented through the boundary conditions file by varying the total pressure (variable **PTOT**, as defined in the *ADPAC07* User's Manual [28]) distribution in the *ADPAC07* **INLETT** boundary data statement.

9.1.4 Solution Procedure

Solutions using the *ADPAC07* code were performed for increasing back pressures from a choke operating point to a near-stall operating point for all four scenarios. Each solution was run at 100% design corrected speed for the AE3007 fan rotor. Enough calculations were performed to define a constant speed operating line for each case. Each solution point was run from a cold (uniform flow) initial condition. The solutions were started

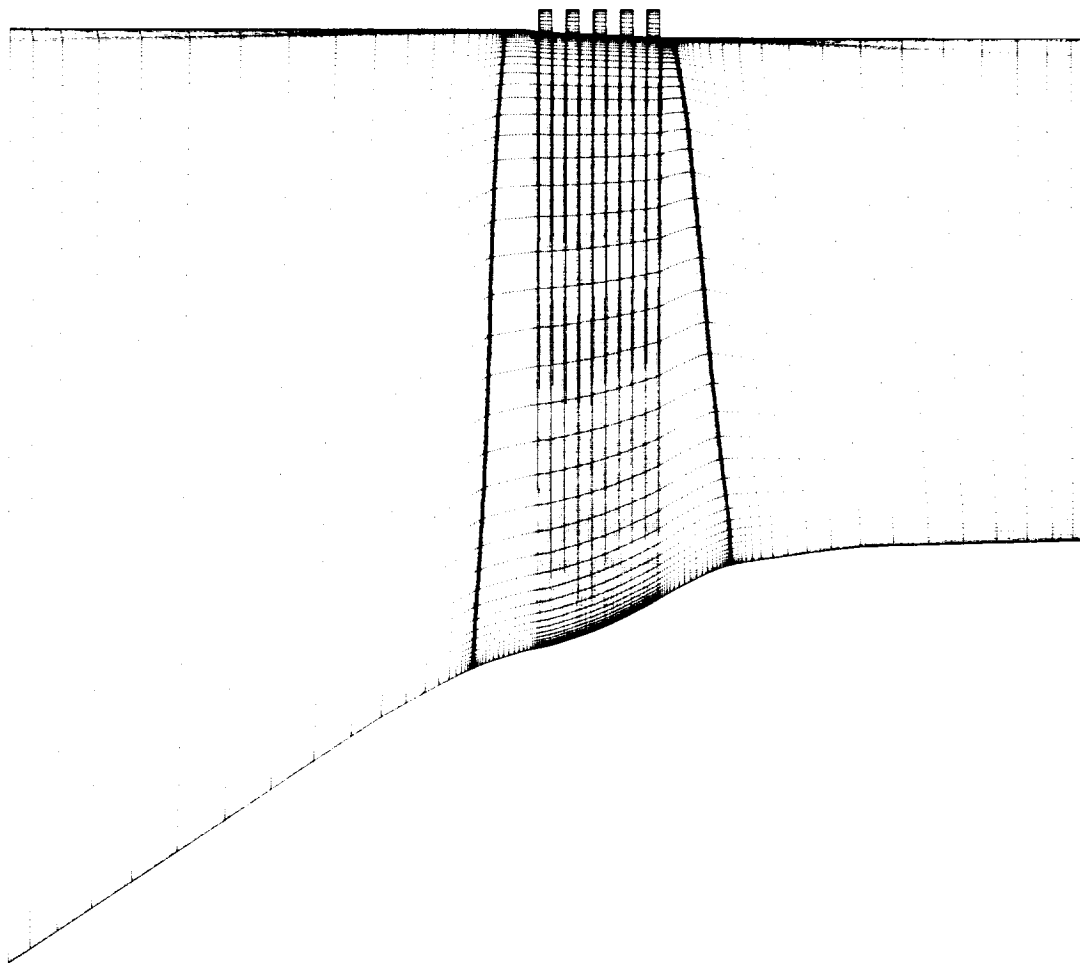


Figure 9.2: Meridional projection of H-type mesh for AE3007 fan rotor with circumferential groove endwall treatment used for radial distortion study.

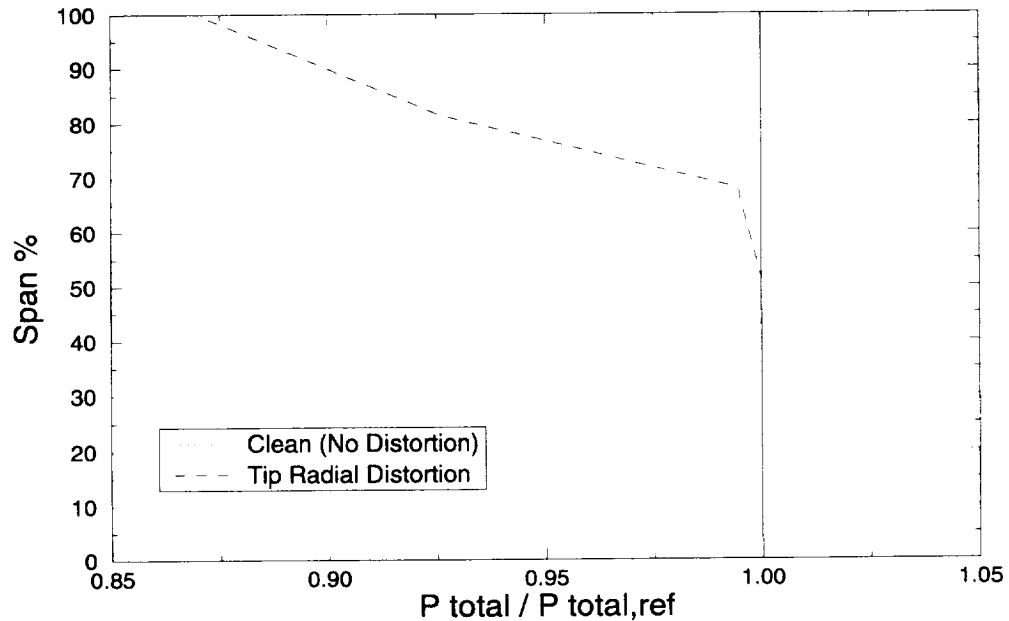


Figure 9.3: Total pressure inlet profiles.

by using the full multi-grid capability in the *ADPAC07* code. The solution was initially advanced on the two coarsest levels of multi-grid for the fine mesh (only one level for the coarse mesh). Following this, the solution was then continued on the fine mesh using multi-grid and local time-stepping with a CFL number (time step multiplier, see input variable **CFL** in the *ADPAC07* User's Manual [28]) of 5. However, as the stall point for each case was approached, the large time steps and multi-grid acceleration scheme appeared to cause some numerical stability problems within the tip clearance grid. At this point, after approximately 250 iteration on the fine mesh, the steady-state CFL number was lowered to 2 and the multi-grid option was disabled, and the solution was restarted. This approach eased the stability problems and allowed for smoother convergence near the stall limit operating point.

In addition to the values of the maximum and RMS residual errors, the values for inlet and outlet mass flows, pressure ratio, and efficiency were output to the convergence file after each iteration. These data outlined the progression of the solution convergence and was found to be a more realistic indicator for final solution convergence.

9.1.5 *ADPAC07* Results and Discussion

The 100% constant speed lines generated from the fine mesh data are shown in the form of pressure ratio versus mass flow and efficiency versus mass flow curves in Figure 9.4. The pressure ratio and efficiency were calculated by comparing mass-averaged conditions at one-half chord behind the trailing edge to the mass-averaged inlet conditions. In addition to the *ADPAC07* results, experimental data (for an untreated rotor) are shown for the baseline AE3007 fan subjected to inflow with and without radial distortion. Results from the *DAWES* [35] flow prediction code are also shown for comparison. A significant difference

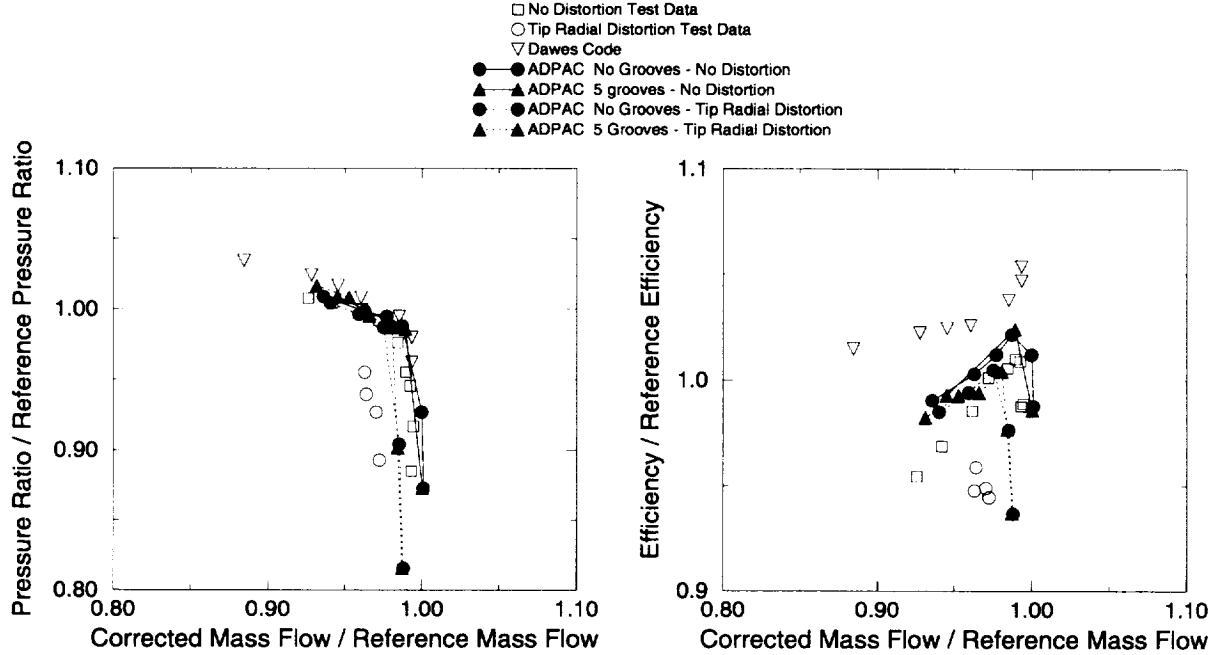


Figure 9.4: Pressure ratio and efficiency for the AE3007 fan rotor.

between the *ADPAC07* results and the *DAWES* code results is apparent, and has been observed in similar comparisons for other geometries. The significant overprediction of efficiency by the *DAWES* code is suspected to be due to a problem in the *DAWES* code wall function formulation.

The fine mesh *ADPAC07* non-distortion results agree well with the corresponding experimental pressure ratio and only slightly over-predict the adiabatic efficiency. The inlet distortion predictions clearly show a decrease in choked mass flow. Difference between the predictions with and without casing treatment are not obvious from these plots. For comparison purposes, a set of four operating points on the speed line was chosen to compare the solutions in more detail: a peak efficiency point and a near-stall point for cases with and without tip radial inlet distortion. Radial distributions of flow properties immediately behind the trailing edge may provide some insight into the differences between the two geometries and how the distorted flow is altered.

Radial distributions of total pressure ratio, total temperature ratio, efficiency and swirl velocity (V_θ) are shown for the various combinations of treated versus untreated rotor, and with and without tip radial distortion at the peak efficiency operating point in Figure 9.5.

AE3007 Fan Radial Distortion Study

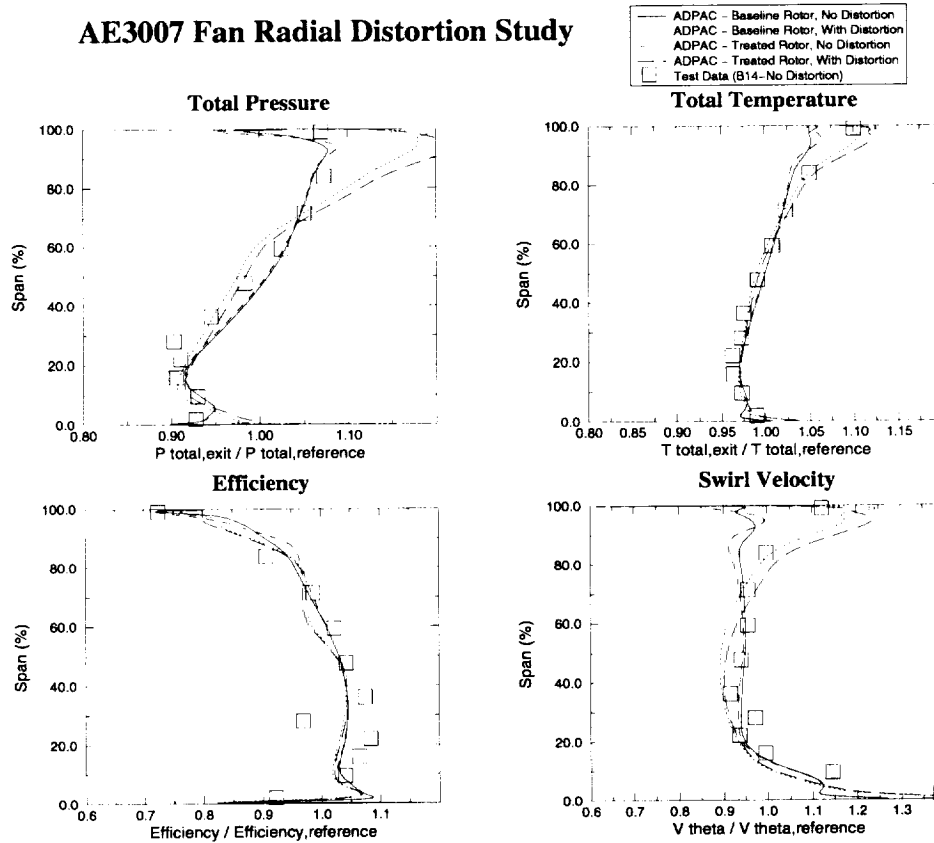


Figure 9.5: Comparison of predicted and experimental radial distributions of total pressure ratio, total temperature ratio, efficiency, and swirl velocity for the AE3007 fan both with and without circumferential groove endwall treatment, and with and without tip radial distortion at peak efficiency.

The radial distributions were obtained by circumferentially-averaging predicted data at an axial station one-quarter chord downstream of the trailing edge. The predicted results show good agreement with the experimental results for an untreated rotor with both a clean inlet and tip radial distortion. Predictions for the geometry with five groove casing treatment did not show any significant differences when compared to predictions for the baseline geometry, except very near the tip, where the greatest differences were expected to occur. The application of the treatment in all cases tested caused an increase in total pressure at approximately 95% span when compared to the untreated rotor. This effect has also been observed in a number of experimental studies involving similar treatments [1]. It must also be noted that the circumferential groove treatment does not cause any appreciable change in efficiency in this application.

A similar set of plots is given in Figure 9.6 for operation at a near-stall condition. In this case, no test data were available for the radially distorted inflow. The effect of the treatment is again observed as an increase in total pressure at roughly 95% span.

Another flow characteristic worth examining is the blade passage shock structure. Plots

AE3007 Fan Radial Distortion Study

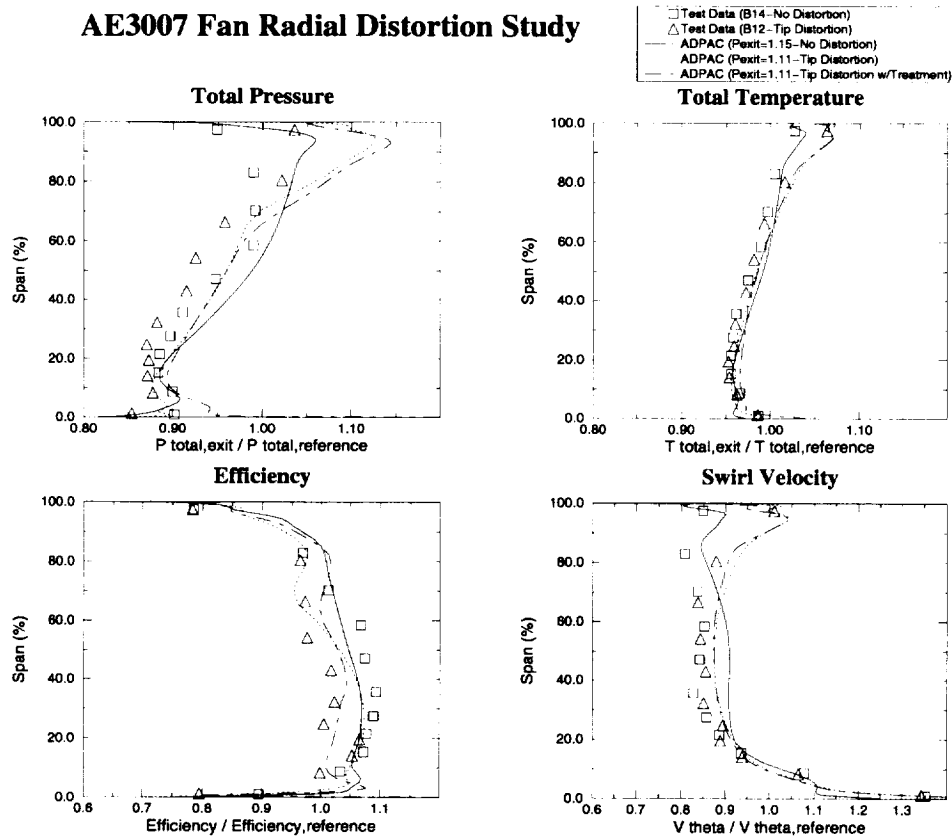


Figure 9.6: Comparison of predicted and experimental radial distributions of total pressure ratio, total temperature ratio, efficiency, and swirl velocity for the AE3007 fan both with and without circumferential groove endwall treatment, and with and without tip radial distortion at near-stall.

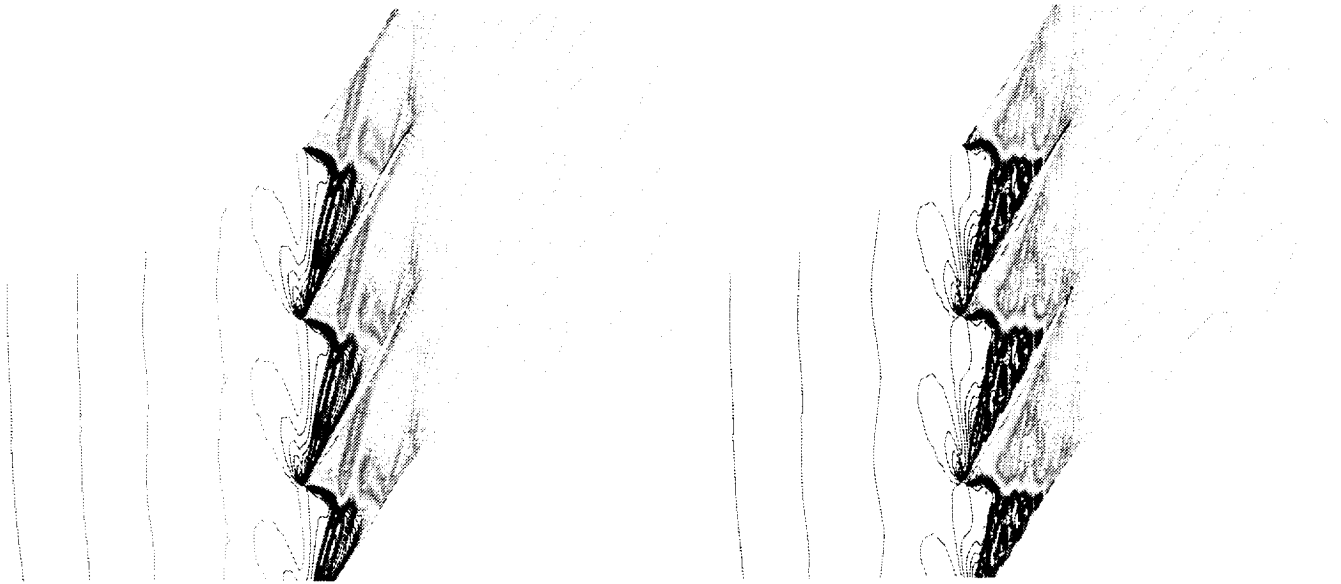


Figure 9.7: Relative Mach number contours located at the tip radial plane at peak efficiency and no inlet distortion for both the standard casing (upper) and endwall treated (lower) configurations.

of predicted relative Mach number contours on a blade-to-blade mesh surface were generated at a radial location corresponding to the blade tip. A series of these plots are given in Figures 9.7, 9.8, 9.9, and 9.10 for both the baseline and treated geometries for the cases of no distortion and peak efficiency, no distortion and near-stall, tip radial distortion and peak efficiency, and tip radial distortion and near-stall, respectively.

Several interesting observations from the series of relative Mach number plots at the blade tip were noted. For each individual geometry, as the rotor approaches the stall-limit operating points, the shock was observed to move forward in the blade passage, and eventually becomes detached from the blade leading edge. Since the radial slices were taken at the physical tip of the fan rotor, the large influence of the tip clearance flow vortex can be seen as the clearance flow passes downstream from the tip and interacts with the passage shock wave. In a comparison of the Mach number plots between the baseline and casing treatment geometries, the effects of the 5 groove casing treatment can be seen as vertical *streaks* in the contour plots aligned with the placement of the grooves. One of the most significant differences between the two geometries at a given flow condition appears at the near-stall operating point with tip radial inlet distortion (Figure 9.10). The location and shape of the shock appears to show the shock detached from the blade leading edge for both geometries at this operating point; however, the solution for the 5 groove casing treatment indicates that the shock forward movement has been retarded by the treatment and the shock position is more stably maintained farther aft in the blade passage. This shock structure may, in part, account for the operation of the casing treatment geometry at slightly higher back pressures (and hence, lower mass flow rates) near stall than the baseline (no groove) geometry.

Comparisons were also performed for the circumferentially-averaged meridional velocities in the blade clearance region for each of the operating conditions described above. The corresponding meridional velocity vector plots are shown in Figures 9.11, 9.12, 9.13,

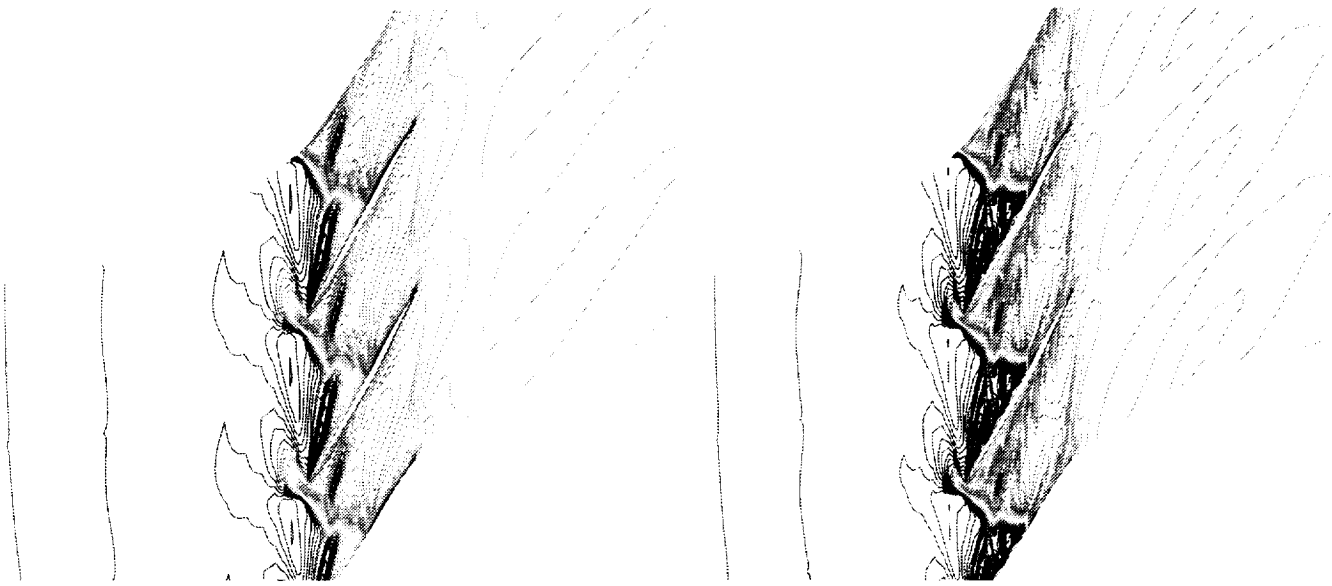


Figure 9.8: Relative Mach number contours located at the tip radial plane at near-stall and no inlet distortion for both the standard casing (upper) and endwall treated (lower) configurations.

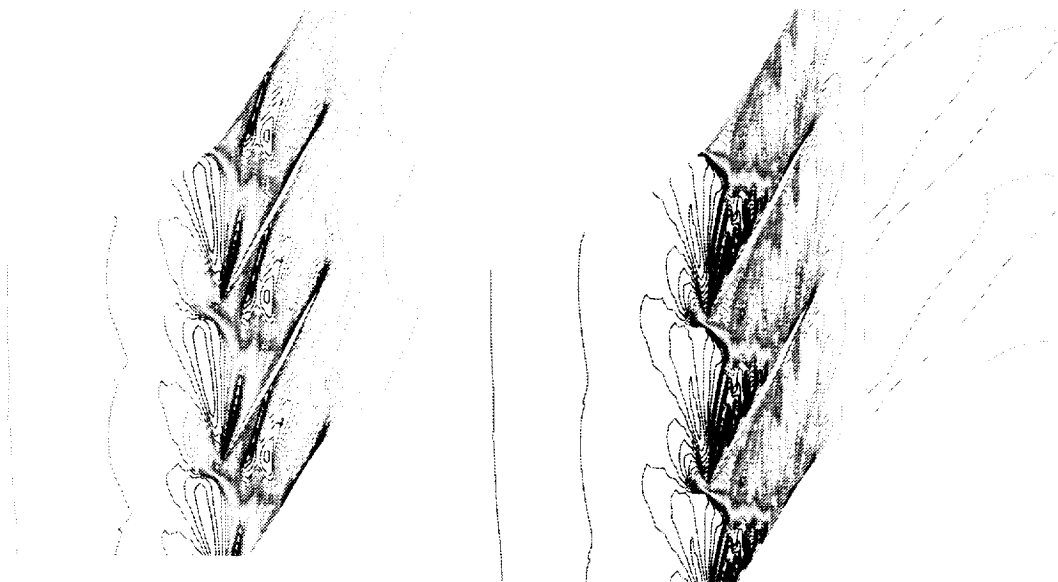


Figure 9.9: Relative Mach number contours located at the tip radial plane at peak efficiency with tip radial distortion for both the standard casing (upper) and endwall treated (lower) configurations.

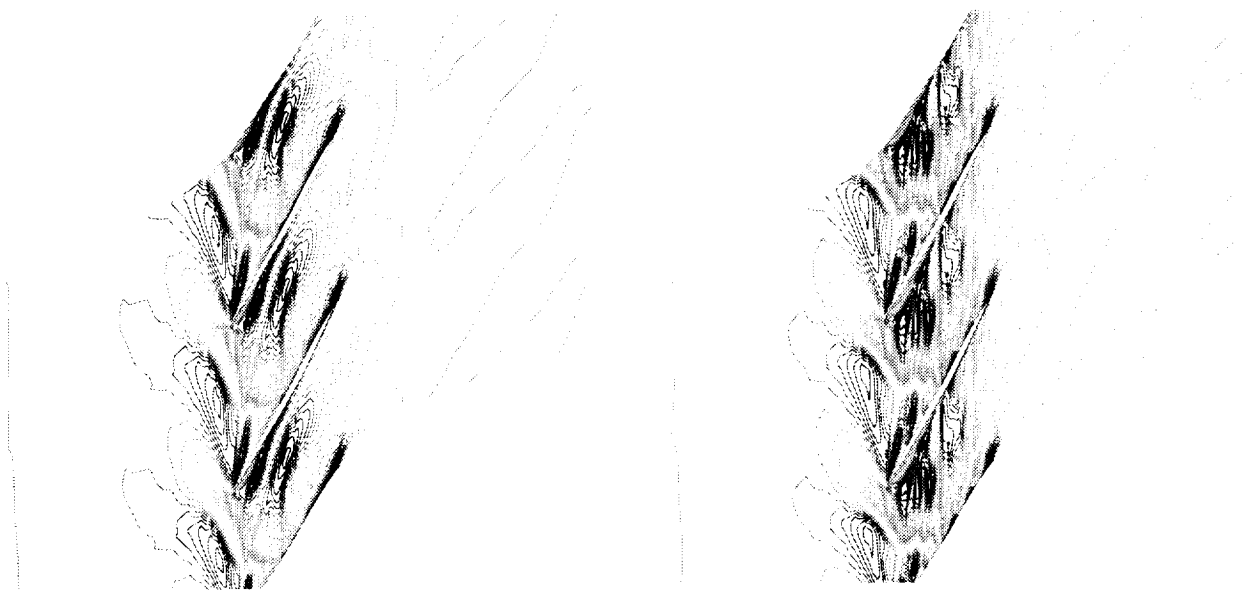


Figure 9.10: Relative Mach number contours located at the tip radial plane at near-stall with tip radial distortion for both the standard casing (upper) and endwall treated (lower) configurations.

and 9.14, respectively. The range of vector origin locations on these figures is from the blade leading edge to the trailing edge in the axial direction and from 99% blade span to the endwall surface in the radial direction. Only every other vector in the main blade passage was plotted for clarity, but all of the vectors in the grooves were plotted in order to maintain the needed resolution. As expected, the vectors suggest small vortical structures generated within each of the 5 grooves compared to a single, larger vortical structure for the baseline geometry at each operating point. The strength and location of these vortices vary with the location of the shock near the tip. For the casing treatment solutions, the groove with the largest vortex correlates to the location of the shock shown in the plots of relative Mach number (Figures 9.7-9.10).

9.1.6 Grid Density Sensitivity

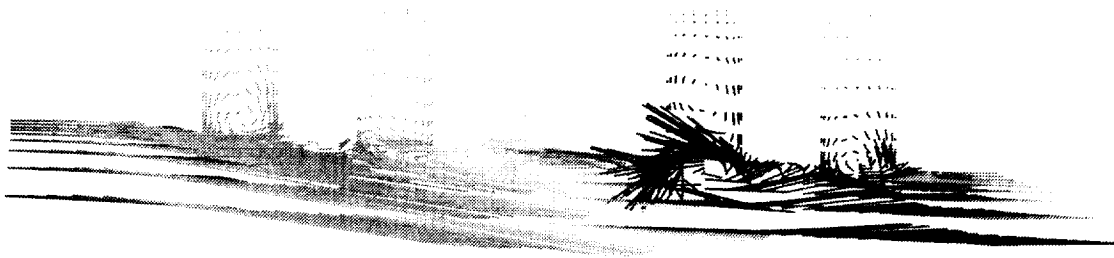
Solutions for the AE3007 fan rotor were carried out on two grids of different grid density. The coarse grid was obtained by removing every other grid line in the fine mesh resulting in a grid with dimensions: 103 x 29 x 25 for the blade passage mesh, 79 x 5 x 7 for the clearance mesh, and 7 x 7 x 31 for each casing treatment groove mesh. The comparison between the *fine* grid and *coarse* grid solutions can be seen in Figure 9.15. The filled symbols represent fine grid solutions, and the open symbols represent coarse grid solutions. Figure 9.15 shows the speedlines for the fan rotor for all four testing scenarios. From these plots, it appears that the coarse grid predicted pressure ratio across the fan agrees well with the fine grid values. However, predicted efficiencies are higher for the coarse grid than the fine grid. This may be due in part to the inability of the coarse grid to adequately define the losses through the blade passage.

As the grid is refined, these losses are more accurately resolved resulting in a more accurate prediction of efficiency. This may also account for the over-prediction of the efficiency values compared with the experimental test data shown earlier in Figure 9.4.

Tip Clearance Region
Peak Efficiency, No Inlet Distortion



No Groove



5 Groove

Figure 9.11: Circumferentially-averaged velocity vectors (shaded by pressure) for peak efficiency operating point without distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span.

Tip Clearance Region
Near-Stall, No Inlet Distortion



No Groove



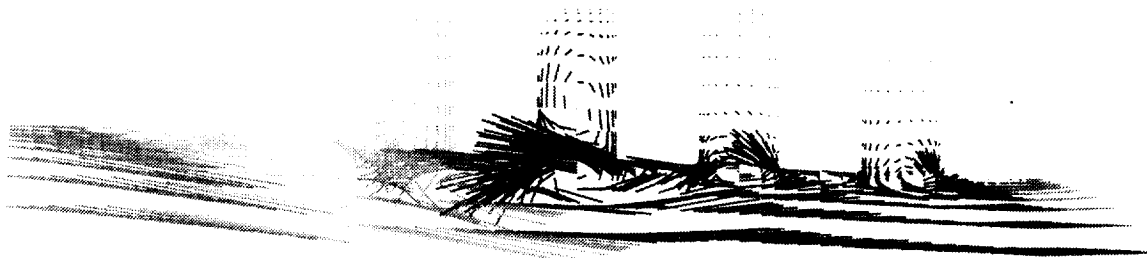
5 Groove

Figure 9.12: Circumferentially-averaged velocity vectors (shaded by pressure) for near-stall operating point without distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span.

**Tip Clearance Region
Peak Efficiency, Tip Radial Distortion**



No Groove



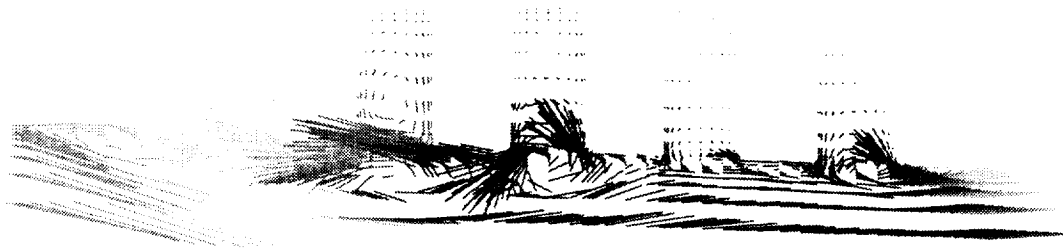
5 Groove

Figure 9.13: Circumferentially-averaged velocity vectors (shaded by pressure) for peak efficiency operating point with inlet distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span.

Tip Clearance Region
Near-Stall, Tip Radial Distortion



No Groove



5 Groove

Figure 9.14: Circumferentially-averaged velocity vectors (shaded by pressure) for near-stall operating point with inlet distortion. Figure shows region from blade leading edge to blade trailing edge and from 97% to 103% radial span.

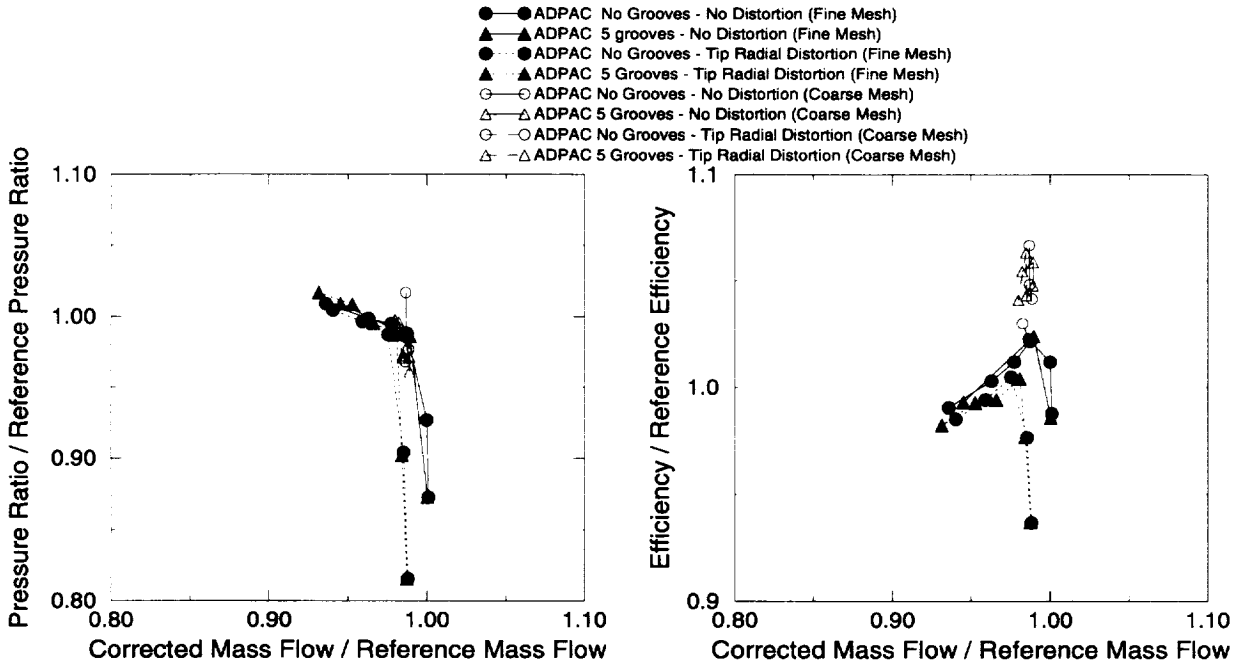


Figure 9.15: Pressure ratio and efficiency speedlines (100% corrected speed) showing comparison between two grid densities. Filled symbols represent *fine* grid solutions, open symbols represent *coarse* grid solutions.

If the fine mesh were further refined, perhaps the predicted values would agree with the experimental data in the limit of the refinement.

Another difficulty with the coarse grid solutions was convergence. Solutions based on the coarse grids tended to require more iterations to reach a converged steady state. As the back pressure was increased, moving the operating point closer to the stall-limit point, the predicted mass flow through the blade passage showed a continual decline and often did not achieve a steady state condition. The mass flow trend as the back pressure was increased from the peak-efficiency point to the near-stall point, as determined from the fine grid predictions, was that the mass flow remained above 95% of the choked mass flow until an additional increase in back pressure pushed the rotor into stall. At this point, the predicted mass flow demonstrated a slow, continual decline versus iteration count. The points where this decline occurred were assumed to be beyond the stall limit operating point and are not shown in the figure. Determining the stall point for the rotor from the coarse grid solution was even more difficult. Determining whether the reduction in mass flow was a result of the fan stalling or a result of the grid resolution reacting to the near-stall operating point was nearly impossible and eventually the coarse mesh solutions were abandoned as unusable.

It does appear, however, that the resolution of the fine grid was adequate for this study. Whereas the coarse grid allowed for a quicker approximation of the pressure ratio, the fine grid was required to adequately resolve the efficiency and, more importantly, determine the near-stall aerodynamic character of the AE3007 fan rotor. Computational limitations and program schedule prohibited examining the effects of employing finer meshes than those described in this study.

Chapter 10

TIME-DEPENDENT CASING TREATMENT ANALYSIS WITH INLET DISTORTION

In this chapter, a series of predictions for a modern turbofan engine fan rotor operating in the presence of circumferential inlet flow distortion is described. The motivation behind these analyses was to examine in detail the aerodynamic impact of circumferential flow distortion on the fan rotor, and to deduce what effect endwall treatments may have in offsetting the adverse impact of circumferential distortion.

10.1 AE3007 Fan Rotor Circumferential Flow Distortion Analysis

The geometry selected for this study was again the AE3007 fan rotor. Details of the AE3007 fan are given in previous sections. The analysis was applied to a baseline rotor, and a rotor with circumferential groove endwall treatment as illustrated in Figure 9.1. The geometry is identical to the geometry used for the radial flow distortion calculations described in the previous chapter, although different mesh systems were utilized for reasons to be discussed later.

In order to model fan performance with circumferential inlet distortion, a time-dependent analysis was performed in order to assess the aerodynamic interaction between the stationary, circumferentially varying distortion pattern and the rotating fan. The most common distortion pattern encountered by most larger turbofan engine fan rotors in operation is described by a once per revolution inlet flow defect resulting from operation at angle of attack or installation effects. The likelihood of multiple distortion cells per revolution is less likely, although engine manufacturers commonly test for the effects of these more complex patterns. For the purposes of numerical analysis, the once per revolution distortion pattern is computationally much more difficult and expensive to perform because the entire fan rotor must be modeled to accurately capture the fan rotor/inlet distortion aerodynamic interaction. This type of calculation is described more fully in the next section dealing with the full engine fan simulation. For the purposes of examining the effects of circumferential distortion on a smaller computational problem, the distortion pattern was modified to appear as a four per revolution defect in inlet flow. This requires that only one fourth, or 6

blade passages of the 24-bladed fan rotor must be modeled. Unfortunately, reducing the problem in this manner has some drawbacks. Figure 2.6 illustrates the effects of varying the distortion pattern in the manner described. It would appear that as the circumferential extent of a single distortion region is reduced, the detrimental effects on fan performance fall off rapidly for circumferential extents less than 45 degrees. In addition, for a given total circumferential extent of multiple distortion patterns, the greater the number of distortion cells, the less impact the distortion has on fan performance. Thus, two 45 degree cells have less impact than a single 90 degree cell, and so on. Regretfully, time and computational costs prohibited a detailed study of these configurations. A general criteria for quasi-steady compressor response to circumferential inlet flow distortion was described by Longley [82]. The criteria is formulated as

$$\frac{1}{4}\lambda > 2\frac{2\pi}{N} \quad (10.1)$$

where λ is the circumferential period of the circumferential distortion and N is the number of blades (hence $\frac{2\pi}{N}$ is the blade pitch). This criteria suggests that for quasi-steady response, the blade of interest and the two nearest neighboring blades operate under similar inflow conditions. For the proposed case, with a blade pitch of 15 degrees, the minimum circumferential distortion pattern based on Longley's criteria would have a circumferential extent of 120 degrees (3 per revolution). In spite of the known drawbacks, the four per revolution distortion pattern (circumferential extent of 90 degrees) was selected as a reasonable compromise between computational cost and aerodynamic impact and importance.

A standard practice in describing inlet distortion patterns for turbofan engine manufacturers is to compute the circumferential distortion index (CDI). The derivation and basis for the CDI is described in Reference [18]. Simply stated, the CDI is defined as:

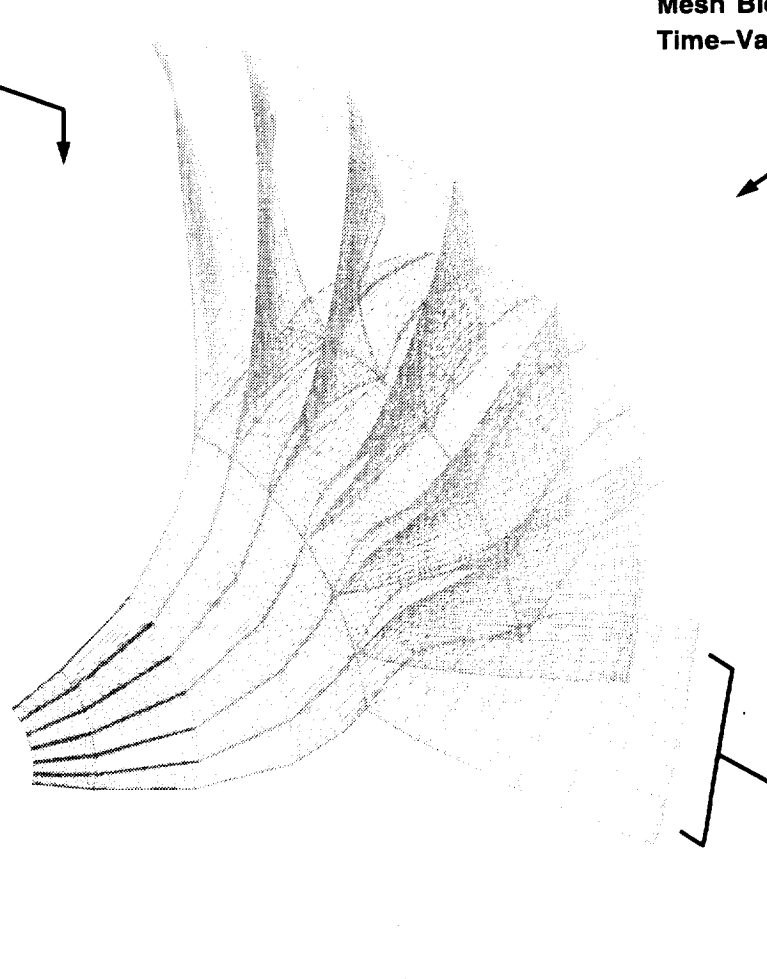
$$CDI = \frac{\Delta P_c}{P} = \frac{P_{average} - P_{min}}{P_{average}} \quad (10.2)$$

where $P_{average}$ is the average inlet total pressure and P_{min} is the minimum inlet total pressure (presumably due to distortion). Various weighting factors are also used to account for multiple distortion lobes per revolution. Multiple lobe circumferential distortion test data were available for the AE3007 fan with CDI values ranging from 0.017 to 0.0219, and therefore, a CDI value of 0.02 was selected for the present set of calculations. Based on the four per revolution distortion pattern, this suggests that the total pressure in the distorted region should be 0.9608 times the inlet total pressure for the undistorted flow (see Reference [18] for details on this calculation).

The calculation was performed by constructing meshes capable of maintaining the stationary upstream distortion pattern as well as accurately modeling the six rotor blade passages. To achieve this end, a sliding interface was utilized on a nearly constant axial surface between the inflow meshes and blade passage meshes across which time and space dependent aerodynamic data were transferred using the *ADPAC07 BCPRR* boundary condition. An illustrative block diagram of the circumferential flow distortion mesh system is given in Figure 10.1. The distortion/rotor aerodynamic interaction problem was therefore treated in the same manner as the rotor/stator interaction problems previously described in this report. A stationary set of grids is coupled to a rotating set of grids with a sliding interface described by a surface of revolution which is very nearly a constant axial plane. The stationary and rotating grid segments have a common circumferential extent, such that spatial periodicity may be applied across both the stationary and rotating mesh segments. In this case, of course, the stationary mesh segments are simply employed to

**Stationary Inlet
Blocks Hold Fixed
Circumferential
Distortion Pattern**

**Rotating Blade Passage
Mesh Blocks See a
Time-Varying Inflow**



**Outline of Circumferential
Groove Casing Treatment
Mesh Blocks
(Removed for Baseline
Calculation)**

Figure 10.1: Multiple-block mesh system for AE3007 circumferential inlet flow distortion study.

define the circumferentially varying (but stationary) inflow distortion pattern, while the meshes discretizing the rotating fan sweep through the convected distortion profile.

10.1.1 Mesh Density Study

In order to minimize the computational resources required for this rather large simulation, a mesh density study was performed to determine the minimum mesh size required to accurately model the AE3007 fan rotor flowfield. Several multiple-block H-type meshes were generated for the AE3007 fan blade with a 5 circumferential groove casing treatment using the TIGG3D grid generation program. A constant speed operating characteristic was predicted for the nominal fan without casing treatment with uniform (steady-state) inlet conditions to evaluate a 105,000 grid point mesh. Solutions for the lightly loaded (low pressure ratio) operating points converged well with residuals dropping over 3 orders of magnitude and the integrated mass flow and efficiency eventually settling to constant values. However, as the back pressure was increased toward the stall point, the integrated flow and efficiency of the fan would drift slowly (over several thousand iterations) toward stall. Although this drift behavior appeared to follow the expected operating characteristic, the residual history was “bottomed-out”, making it difficult to determine when the solution was completed. Inadequate resolution of the shock/tip-vortex interaction near the blade tip was believed to be the cause of this undesirable convergence behavior. Satisfactory solutions near the stall point were eventually obtained with a mesh of approximately 142,000 grid points.

Steady-state calculations without distortion were performed using the selected mesh with the 5 circumferential groove casing treatment (152,000 total points). These solutions were used to assess the casing treatment grid resolution and the effect of the grooves for an undistorted inlet flow. Figures 10.2 and 10.3 compare predicted and experimental pressure ratio versus mass flow and efficiency versus mass flow, respectively, for the stable solutions that were obtained for conditions ranging from choke to peak efficiency for both the treated and non-treated rotor geometries using the 142,000 and 152,000 point meshes, respectively. These solutions indicated that the H-type grids used to resolve each groove (7 axial, 7 radial, 41 tangential) were adequate. The comparison of the treated and non-treated rotor geometries for these solution indicated that the 5 groove treatment does not *significantly* affect the overall performance of this fan with an undistorted inlet flow. The results are essentially identical to those obtained during the radial flow distortion study described in the previous chapter.

10.1.2 Parallel Performance Study

Several portions of the time-dependent calculations for the circumferential distortion study were performed on the NASA-Lewis LACE (Lewis Advanced Cluster Environment) computing platform. The LACE cluster consists of a networked set of IBM RS-6000 workstations, as well as a self contained multiprocessor workstation referred to as the SP1. Prior to executing the full analysis, a study of parallel computing performance was performed. This study examined variations in computational speed based on number of parallel processors, boundary condition type, and load balance. Load balance was accomplished by specifying the block/processor assignment via the *casename.blkproc* file (see *ADPAC'07 User's Manual* [28] for details). Table 10.1 lists the various configurations tested and the resulting CPU time per iteration for the LACE SP1 cluster. The CPU timing studies were per-

AE 3007 Type III Fan Casing Treatment Analysis

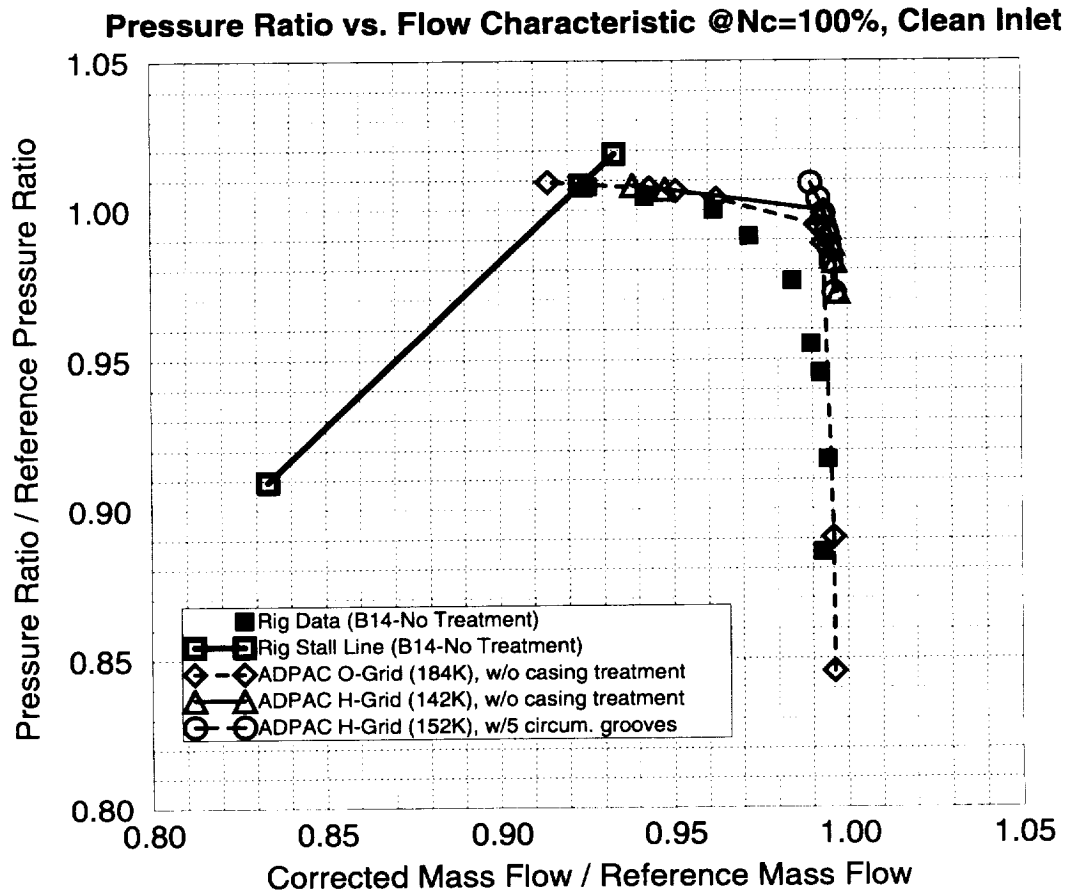


Figure 10.2: Comparison of predicted and experimental pressure ratio versus mass flow characteristic for the AE3007 fan (100% corrected speed).

AE3007 Type III Fan Casing Treatment Analysis

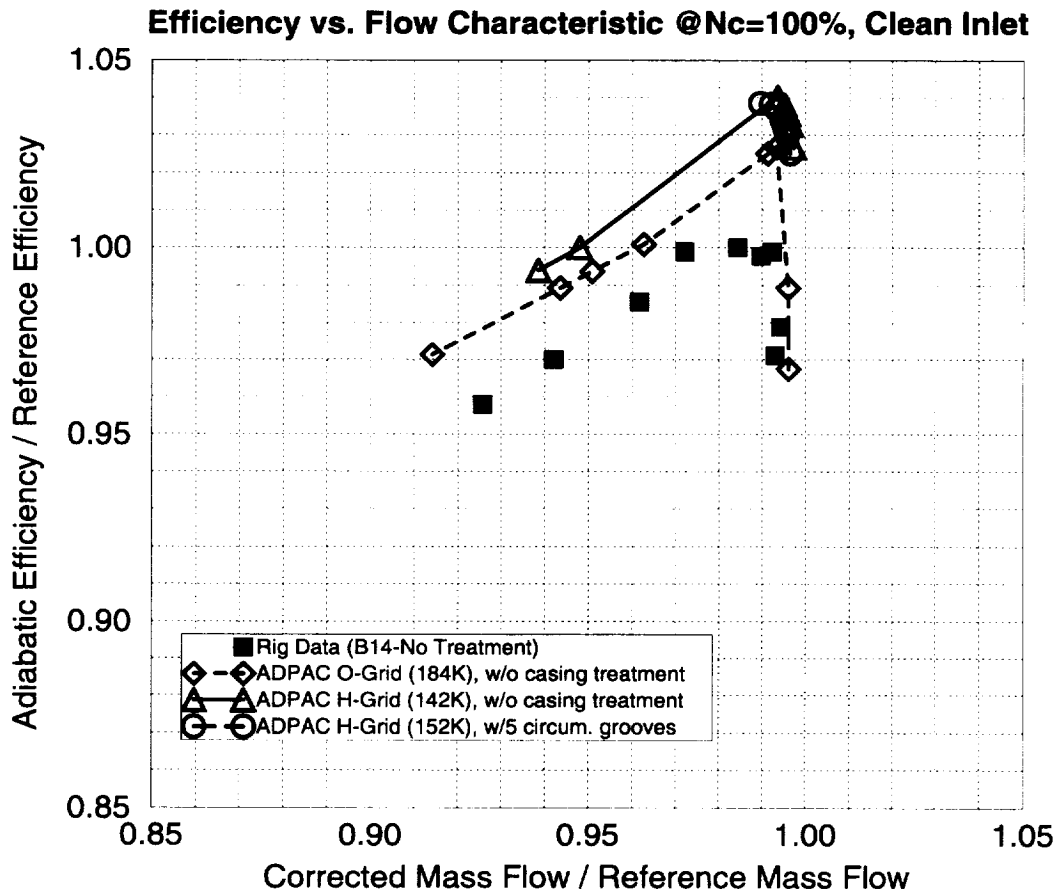


Figure 10.3: Comparison of predicted and experimental efficiency versus mass flow characteristic for the AE3007 fan (100% corrected speed).

**Summary of ADPAC Parallel Computing Performance on LACE SP1 Workstation
Cluster for Circumferential Distortion Study Untreated Rotor Mesh**

# Processors	Block/Processor Assignment	CPU sec./iteration	Communication/Boundary Application sec./iteration	
6	ADPAC Default	167.0	--	(BCPRR)
6	Blade Passage	37.9	8.9	(BCPRR)
8	Hand-Specified	46.2	18.0	(BCPRR)
6	Blade Passage	28.6	3.4	(BCPRM)
8	Hand-Specified	38.6	14.6	(BCPRM)

- ADPAC Default block/processor assignment is in incremental order (block 1 on proc 0, block 2 on proc 1, etc.)
- Blade Passage block/processor assignment stores all 3 blocks associated with a given blade passage on a single processor (blocks 1,2,and 3 on proc 0, etc.)
- Hand-Specified block/processor assignment attempts to load balance for 8 processors (blocks 1, 3, 4, 6, 5, and 9 on proc 0, blocks 10, 12, 13, 15, 16, and 18 on proc 1, and blocks 2, 5, 8, 11, 14, and 17 on procs 2, 3, 4, 5, 6, and 7, respectively).

Table 10.1: Summary of *ADPAC07* parallel CPU performance on LACE SP1 cluster for circumferential distortion study untreated rotor mesh.

formed on the mesh representing the untreated rotor, with every other mesh line removed to permit relatively small CPU time per iteration. The results were assumed to scale proportionally for the fine mesh. The initial run, based on 1 processor, is equivalent to the serial execution speed. All other combinations utilized either 6 or 8 processors. Since the mesh consists of 18 mesh blocks, most of the combinations tested involved 3, 6, and 9 processors. The results presented in Table 10.1 are representative of the overall results. The optimum configuration was to utilize 6 processors, with the mesh blocks distributed such that the group of three mesh blocks associated with a given blade passage were on a common processor. Conceptually, this arrangement gives the best load balance. Compared to the 8 processor configuration, the additional communication overhead does not justify the additional computing power of 2 extra processors. A comparison of the 6 and 8 processor arrangements using both the **BCPRR** and **BCPRM** boundary conditions is also of interest in this study. The **BCPRR** and **BCPRM** boundary specifications perform essentially the same function; that is, these calls perform the time/space aerodynamic data interpolation at the sliding interface used in the rotating/nonrotating mesh configuration. The author's experience suggest that this boundary specification is the primary source of interprocessor communication overhead for this type of calculation. The **BCPRM** routine was constructed to reduce the communication overhead found in multiple **BCPRR** calls. Hence, a single **BCPRM** can perform the same function as several **BCPRR** calls. Employing the **BCPRM** calls for the parallel performance test case resulted in a significant reduction in communication and boundary application time (18.9% to 62.2% based on the number of processors) compared to the **BCPRR** test results. The optimum configuration using 6 processors with "perfect" load balance and **BCPRM** boundary specifications was therefore selected for the time-dependent calculations.

10.1.3 Circumferential Distortion Results

Both the baseline and endwall treated time-dependent calculations were initially advanced in a steady state mode to provide a reasonable starting point for the unsteady simulations. The time-dependent solutions were then initiated from the corresponding steady state solutions. The solutions were advanced in time until a time periodic solution was observed. Approximately 4 periods, or one complete revolution of the wheel were required before a time-periodic response was observed. The time-dependent computations were performed using a variety of computing resources. Both serial and parallel computing strategies were employed at various stages of the calculation as computing resources became available. At no time during any of the simulations was any evidence of rotating stall encountered. Under the present calculation strategy, if the aerodynamic conditions were such that a rotating stall should occur, the analysis should have been capable (in theory) of capturing this important phenomenon. Future predictions employing this strategy should concentrate further on demonstrating this capability.

The time-averaged predicted mass flow rate for both the treated and untreated configurations were essentially identical. Looking at the mass-average of the time-averaged solution, the treated configuration resulted in an exit to inlet total pressure ratio which was 0.44% greater than the untreated configuration. The resulting adiabatic efficiency of the treated configuration was 0.2% lower than the baseline rotor. The analysis of the predicted results focused on examining the aerodynamic response of the rotor when the distorted inflow was encountered. An effort was made to determine the physical mechanisms which counterract the negative effects of the distortion, and to investigate the beneficial effects (if any) of the endwall treatment to alter these reactions (e.g. does the treatment *enhance* the normal rotor response to the distorted flow, or are some other new physical features brought into play?). Predicted instantaneous inflow and outflow total pressure contour plots are given for both the baseline rotor and the treated rotor in Figures 10.4 and 10.5, respectively for what was estimated to be a near stall operating condition (due to the computational expense of the calculation, only a single operating point could be processed). The contours are plotted on a mesh surface which is nearly a constant axial plane. The circumferential distribution of total pressure at the inflow boundary (representing the specified inflow distortion pattern) is essentially a square wave, with clearly defined boundaries between distorted and undistorted flow. Since the distortion pattern is applied to the stationary mesh surfaces, the boundary between distorted and undistorted regions do not necessarily lie along purely radial rays.

The downstream total pressure contours from both the untreated and treated configurations clearly display the four lobed pattern expected from the specified inlet distortion pattern, although the pattern is more apparent in the outer 40% of span than in the inner 60% of span. The total pressure defect at the inflow results in a corresponding reduced total pressure region at the outflow boundary plane. A particularly troubling region of very low total pressure exists near the outer flowpath boundary in the circumferential center of the outflow low total pressure lobe. It is this low total pressure region, resulting from clearance flows and endwall boundary layer buildup, in conjunction with the low total pressure distorted inflow, which leads to eventual stall at higher pressure ratios.

The overall level of deviation in total pressure at the outflow boundary due to inflow distortion is reduced (compared to the deviation at the inlet) by the observation that the rotor reacts to the retarded flow regions with increased blade aerodynamic loading. An illustration of the predicted radial spanwise distribution of circumferentially averaged total

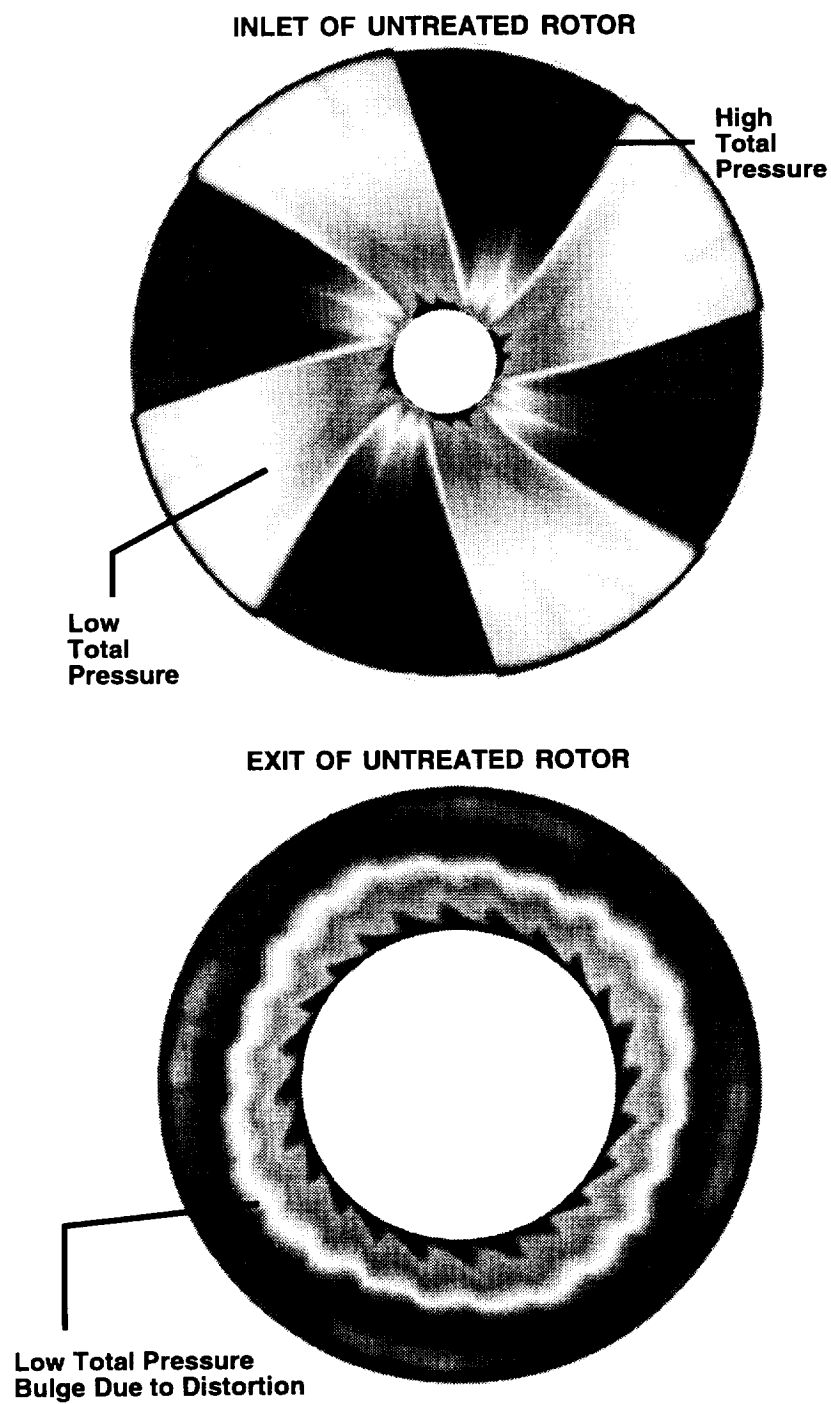


Figure 10.4: Inflow and outflow axial total pressure contour pattern for AE3007 circumferential distortion calculation (untreated rotor, near stall operating point).

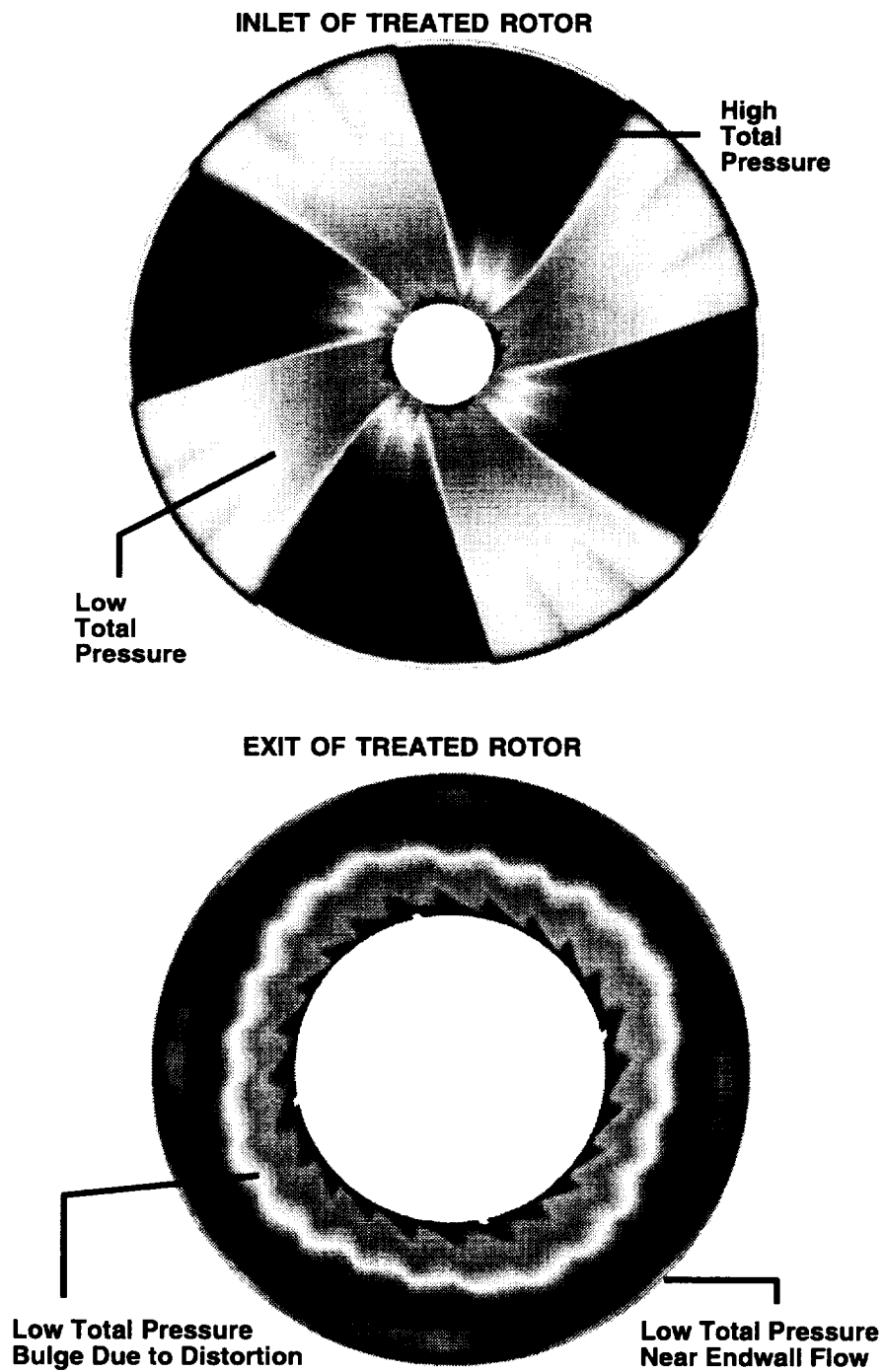


Figure 10.5: Inflow and outflow axial plane total pressure contour pattern for AE3007 circumferential distortion calculation (treated rotor, near stall operating point).

AE3007 Circumferential Inlet Distortion Study

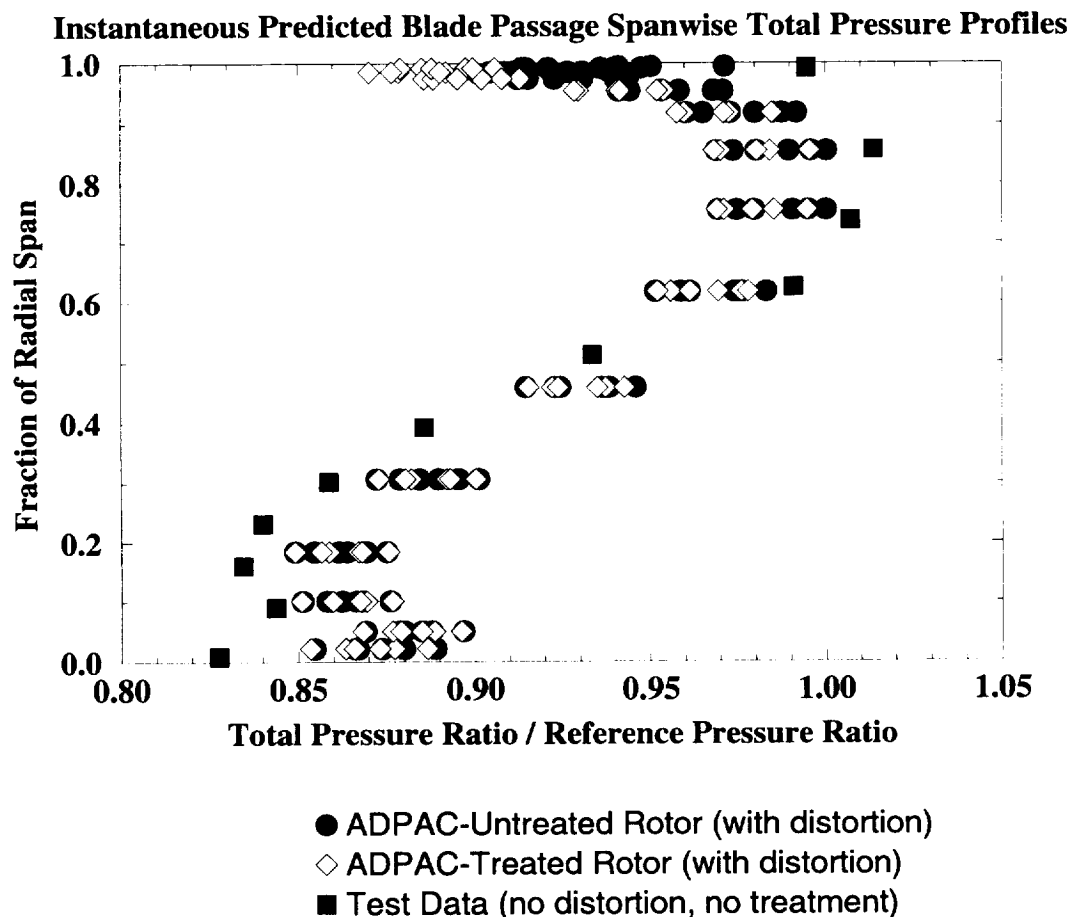


Figure 10.6: Instantaneous predicted radial distribution of circumferentially averaged total pressure profiles for several blade passages of the AE3007 fan rotor (untreated) with circumferential inlet distortion.

pressure is given for an instant in time for each of the six blade passages for both the untreated and treated rotors in Figure 10.6. The corresponding steady state (no distortion) predictions and no distortion test data are also plotted for comparison. The variation in predicted total pressure is essentially centered about the steady state predictions, except in the vicinity of the tip, where both the treated and untreated rotor simulations indicate a significant reduction in total pressure due to the distortion. It is interesting to note that the circumferentially mass-averaged total pressure near the tip of the treated rotor is less than the corresponding values for the untreated rotor, in spite of the fact that the overall pressure ratio was higher for the treated configuration. This suggests a redistribution of flow due to the endwall treatment away from the low total pressure region (near the tip) to higher pressure ratio spanwise positions. This change in streamline distribution due to the recirculating flow of the endwall treatment was also observed for the steady state calculations described in Chapter 7.

Blade loading was observed to change fairly dramatically from passage to passage as the distorted inflow region was traversed. Both the treated and untreated flow predictions show

identical passage-to-passage variations in total pressure ratios from 0% to near 50% span. Beyond 50% radial span, predictions for the treated configuration shows a smaller variation in total pressure ratio variation than the prediction for the untreated rotor. Total pressure ratio very near the endwall is significantly lower for the treated configuration than the untreated configuration. The minimization of the blade-to-blade variation in total pressure ratio can be viewed as a beneficial effect of the endwall treatment for this time-varying flow. When the distorted inflow region is encountered, there is a corresponding increase in blade loading to achieve the desired exit static pressure. This mechanism provides a reasonable explanation for the compressor response to inflow distortion patterns previously described. If, in the average of time, the rotor is operating at a near stall condition, the increase in loading required to traverse the distorted flow and maintain the desired exit static pressure could exceed the stall capability of the airfoil and cause premature stall. If, on the other hand, the circumferential extent of the distortion is small, then it is possible that the rotor can recover from the excessive loading before catastrophic stall occurs. This implies that the "critical" circumferential extent of the distorted region (described in Figure 2.5) is a direct function of the ability of the blade to survive a temporary increase in loading which would otherwise cause stall. Smaller circumferential extents imply a shorter duration for the increased loading, and consequently, the rotor is more likely to recover from the adverse operating condition. It is clear that if the circumferential extent is large enough, the rotor will stall, and no recovery is possible. Additional increases in circumferential extent beyond this critical point will not further degrade the performance. Therefore, anything which permits the blade to survive longer durations of increased loading will enhance the stall characteristics of the rotor. It is also probable, then, that anything which minimizes the variations in the rotor blade loading from the time-averaged state will also provide a beneficial effect in terms of stall. It is possible that endwall treatments provide a means by which the blade can tolerate increased loadings for longer periods of time, and thus improve stall margin. A literature survey of experimental tests for endwall treatments with multiple-lobed circumferential distortion patterns was inconclusive on this point; however, the predictions from this study indicate that this description of the flow phenomena is valid.

Many of the observed characteristics of the time-dependent calculations for the treated rotor were consistent with the characteristics of the steady state characteristics from the radial distortion study. These observations generally imply that the physical mechanisms providing stall enhancement afforded by the use of circumferential grooves is essentially the same for both radial and circumferential distortion patterns. This statement is not necessarily remarkable, but does imply that the stall enhancement features of circumferential grooves results from essentially quasi-steady state flow physics.

No estimate of the improvement in stall margin is possible based on this single simulation. Time constraints prohibited simulating additional operating points which might have otherwise permitted an estimate of stall margin degradation due to circumferential distortion, and stall margin enhancement due to the grooved endwall treatment. Widespread availability of supercomputer performance level workstations will ultimately permit more complete studies of this nature.

Flight Mach Number	0.2
Altitude	Sea Level (ISA)
Corrected Speed	100%
Angle of Attack	30 degrees

Table 10.2: Summary of flight conditions for *ADPAC07* engine/fan section distortion calculation.

10.2 AE3007 Engine Fan Section Inlet Flow Distortion Simulation

This section describes a calculation performed for a realistic turbofan engine inlet/fan section configuration to predict the behavior of a modern fan rotor operating in the presence of inlet flow distortion resulting from angle of attack. The geometry selected for this study was again the Allison AE3007 turbofan engine. A schematic cross section of the engine inlet, flowpath, and primary fan section components is given in Figure 10.7. This analysis was intended to provide some insight into the distortion pattern resulting from a realistic engine inlet operating under severe angle of attack flight conditions and to predict the time-dependent aerodynamic interaction of the resulting distortion pattern with the rotating fan. In addition, the downstream effects of the distortion pattern were also of interest, and therefore special consideration was given to modeling additional downstream components in the engine, and, in particular, the fan section. The modeling of these additional components also provides the added feature of realistically back pressuring the fan rotor in that event that simple radial equilibrium is invalid.

The flight conditions selected for this calculation are listed in Table 10.2. These conditions were selected as being representative of a “worst case” angle of attack/inflow distortion flight condition commonly encountered by modern commercial aircraft.

10.2.1 Mesh Generation

Flow paths for the engine components of the Allison AE3007 fan engine were assembled and combined to create a composite of the two primary flow paths through the engine. Figure 10.7 shows the flow paths and the preliminary grid block boundary definitions generated by the turbomachinery grid generation code *TIGG3D*. With this particular engine configuration, the flow initially is split by the engine cowl into the external flow and the internal engine flow. The internal flow is then drawn into and through the fan rotor and then split into the core and bypass flows. The bypass ratio of the AE3007 engine is 5:1. As shown in Figure 10.7 the core flow passes through the core compressor and exhaust diffuser into the barrel of the combustor. The flow path for the combustor liner is not shown. After exiting the combustor barrel, the core flow passes through the high pressure (HP) and (LP) turbine flow paths and finally exhausts through a mixed flow exit nozzle configuration. The

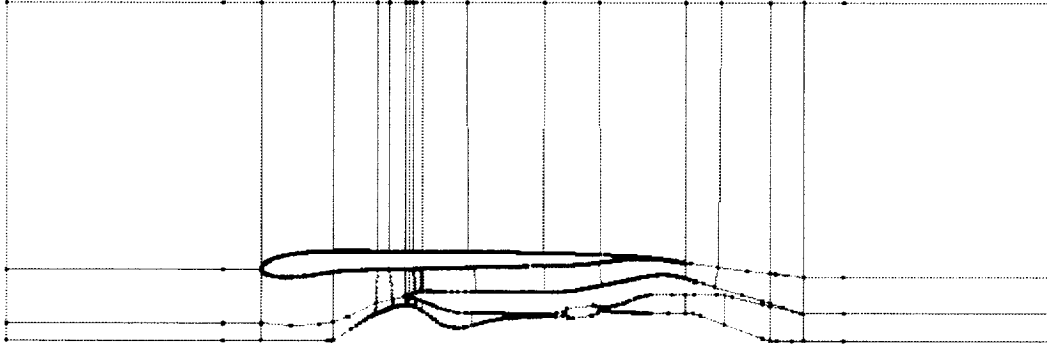


Figure 10.7: AE3007 engine flowpath and *TIGG3D* block boundary illustration.

bypass flow path, on the other hand is relatively simple. After passing over the flow splitter, the bypass flow traverses the length of the engine and is then mixed with the core flow before exiting out of the exhaust nozzle at a design Mach number of approximately 1. The calculation utilized specialized boundary conditions to simulate many of the internal flow components. The calculation core flowpath was terminated following the core inlet guide vane (IGV). The effects of the core compressor, combustor, and HP and LP turbine components were simulated using bulk boundary specifications at the core IGV exit plane and the core nozzle inlet plane. The bulk boundary conditions represented the mass-averaged flow properties of the core flow for the flight condition selected, and consequently, the core flowpath is not gridded between the IGV exit plane and the core flow nozzle inlet plane. The bypass flow was numerically modeled, and therefore the bypass flow path is at least partially gridded.

Before final mesh generation, the *TIGG3D* mesh generation program was modified so that the rows of subblocks in the grid generation process, which in this case delineate the core, bypass, and external flow field, could have different numbers of axial points, and the grid could be output as three separate blocks instead of a single grid. This allowed the number and distribution of axial points in each row to be controlled independently. This produced a more efficient mesh system and did not force the use of wasted points in one block because of geometric constraints, such as a blade row, imposed by another block.

The *TIGG3D* program generates a 3-D grid for one pitch of a chosen blade row in the flow paths defined within the input deck. For this calculation, there are three blade rows defined in the flow paths: the fan rotor with 24 blades, the core vane with 58 blades, and the bypass vane with 74 blades. The *TIGG3D* program was run one time for each blade row. Each run produced a 3-D grid bounded by the flow paths in the meridional plane with a circumferential pitch equal to the pitch of the given blade row (360 degrees divided by the number of blades). Most of the overall grid for the full engine simulation was derived from the grid generated for the fan rotor. The grids for the core vane and the bypass vane were extracted from the subsequent *TIGG3D* runs for each respective blade row.

The full flow field for this calculation is decomposed into 98 grid blocks. The schematic diagram given in Figure 10.8 illustrates how the 98 blocks are distributed for this simulation.

The mesh consisted of 4 “superblocks” which each contained 24 blocks for the fully three-dimensional non-periodic calculation of the fan flow, external flow, bypass exit flow, and core exit flow. The superblocks are labeled in Figure 10.8 as s#1, s#2, s#3, and s#4 respectively. These grids were obtained by duplicating the single pitch, periodic grid

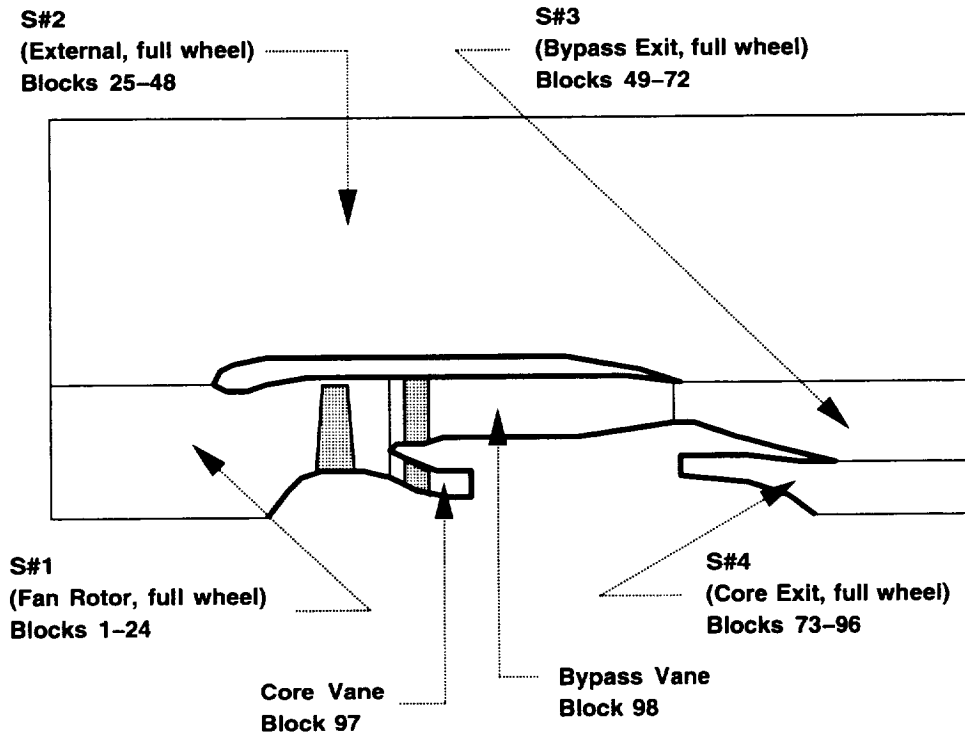


Figure 10.8: AE3007 engine flowpath and *ADPAC07* block numbering illustration.

generated for each of these regions of the flow field by *TIGG3D* for the fan rotor blade row. This produces a ‘full wheel’ grid for the fan rotor, external flow, bypass exit, and core exit. These 4 superblocks account for 4 x 24 or 96 of the 98 blocks in the calculation. Block number 97 is a single pitch 3-D periodic grid for the core vane, which has 58 blades. Block number 98 is a single pitch 3-D periodic grid for the bypass vane, which has 74 blades.

The fan rotor flow field (s#1) begins at the inlet plane and extends axially to the leading edge of the splitter downstream of the rotor. It is coupled to the core flow and bypass flow with a mixing plane boundary condition. The flow variables from all 24 blade passages were circumferentially mass-averaged and transferred as inlet conditions for the core and bypass vane flows. Likewise, the values from the single blade passage calculation of the core and bypass vanes were circumferentially mass-averaged and transferred as exit conditions for all 24 blade passages of the fan rotor flow field. The external flow field (s#2) extends axially from the inlet to the exit, and radially from the cowl to the far outer boundary. It is coupled to the fan rotor and bypass exit flows through usual *ADPAC07 PATCH* block boundary data transfer. The bypass exit (s#3) extends axially from the bypass nozzle discharge to the exit of the flow domain. This region is coupled to the internal bypass flow (labelled bypass vane) by mixing plane boundary conditions at the bypass nozzle discharge. The core exit (s#4) extends axially from the turbine exit to the exit of the flow domain. This region is not directly coupled to the core vane flow in the numerical solution, but instead utilized engine test data at the turbine exit to define inflow parameters. The core vane flow field, which begins at the leading edge of the flow splitter, terminates at the axial location corresponding to the leading edge of the 1st blade row of the core compressor. A static pressure exit condition was specified at the core vane exit plane to simulate the proper core flow.

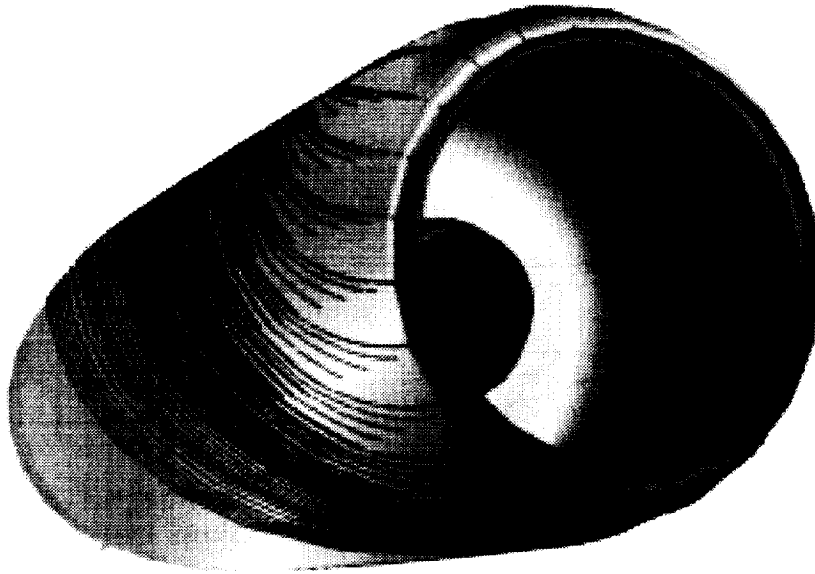


Figure 10.9: Predicted surface static pressure countours and near surface streamlines for the AE3007 fan (Mach=0.2, angle of attack=20.0 degrees).

10.2.2 Mesh Density/Flow Condition Study

Several mesh systems were tested for the full engine fan section distortion study identified for this task. Preliminary meshes were generated for the AE3007 geometry, and initial calculations were performed to analyze the mesh dependence of the computed results. The behavior of the near inlet flowfield at large angles of attack was of particular interest in order to define a significant, yet realistic inlet flow distortion pattern for the fan section calculation. Several inlet-only calculations were performed to assess the degree of inlet distortion at various angles of attack. Figure 10.9 illustrates the predicted surface static pressure contours and near surface streamline flow pattern for an inlet-only (no fan blading present) calculation for the AE3007 engine. The fan mass flow rate was adjusted through internal specification of boundary conditions within the fan. The associated near fan face total pressure contour pattern for each run was plotted and these distortion patterns were examined for various flight conditions in order to select the “best” (most realistic) flight conditions for the final time-dependent calculation. The flight conditions described in Table 10.2 were based on the results of the mesh density/flow condition study.

10.2.3 Results

The time-dependent engine simulation was initiated from a steady state solution on the same mesh calculated at 0 degrees angle of attack. The time-dependent simulation employed the

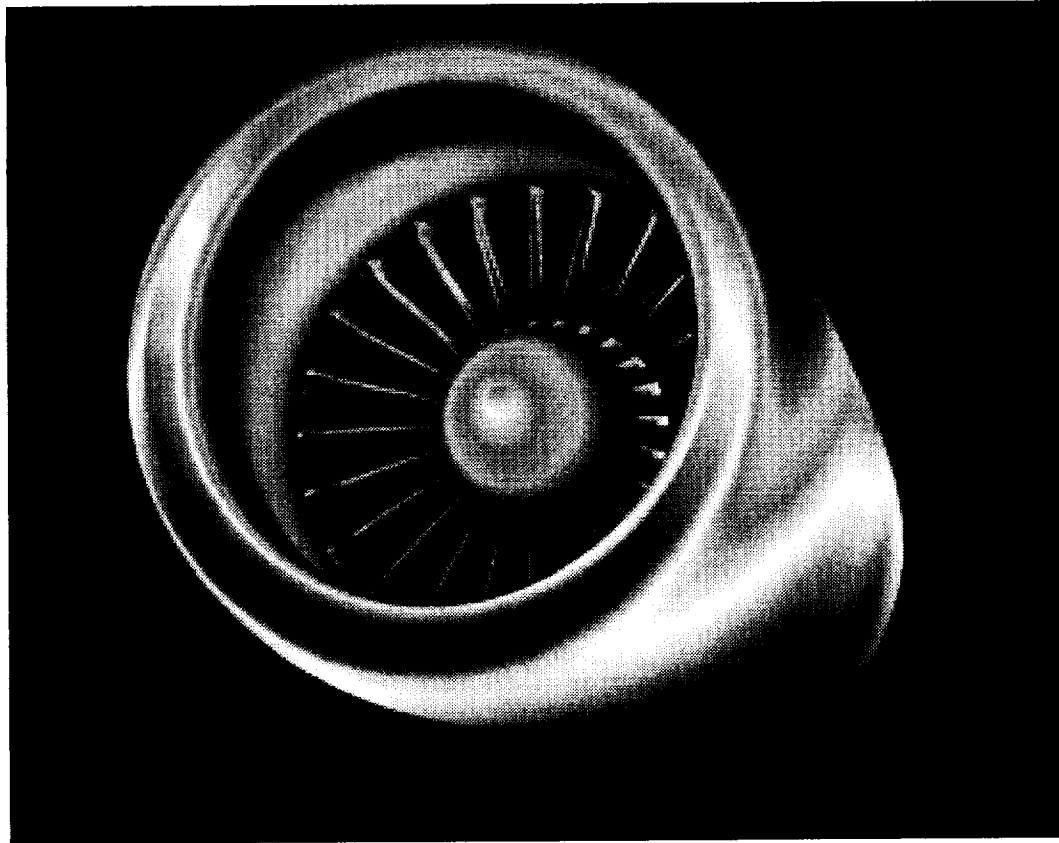


Figure 10.10: Predicted instantaneous surface static pressure contours for AE3007 engine simulation (Mach=0.2, angle of attack=30.0 degrees).

explicit time-marching strategy as the near wall mesh spacings were sufficiently large to permit relatively large explicit time steps. Approximately 10,000 time steps were required for a complete rotation of the fan rotor, and the overall simulation required approximately 2 rotor revolutions to achieve a time-periodic behavior. A sample graphic illustrating the instantaneous engine surface static pressure contours is given in Figure 10.10. The asymmetric pressure pattern on the outer surface of the cowl clearly indicates the effects of the angled inflow.

An illustration of the predicted instantaneous near fan face inlet total pressure contours is given in Figure 10.11.

The contour pattern indicates the nearly annular region of low total pressure near the cowl which represents the cowl boundary layer. The boundary layer thickness is slightly larger on the lower surface due to the angled inflow. The total pressure defect represents at least one distortion aspect of the angle of attack operation. Another aspect is related to swirl flow due to angle of attack. As the inlet captures the angled inflow, some flow straightening occurs as a natural consequence of the inlet bounding surfaces. Unfortunately, along the sides of the inlet (at roughly the 3:00 o'clock and 9:00 o'clock positions viewed looking aft) the flow is not sufficiently straightened, and from the rotor plane of reference, this angled inflow represents local regions of positive and negative bulk swirl, respectively. This bulk swirl effect in effect modifies the incidence angle at the rotor and can have significant impact on fan stall margin (see e.g. [18]). The inlet total pressure contour pattern is not symmetric

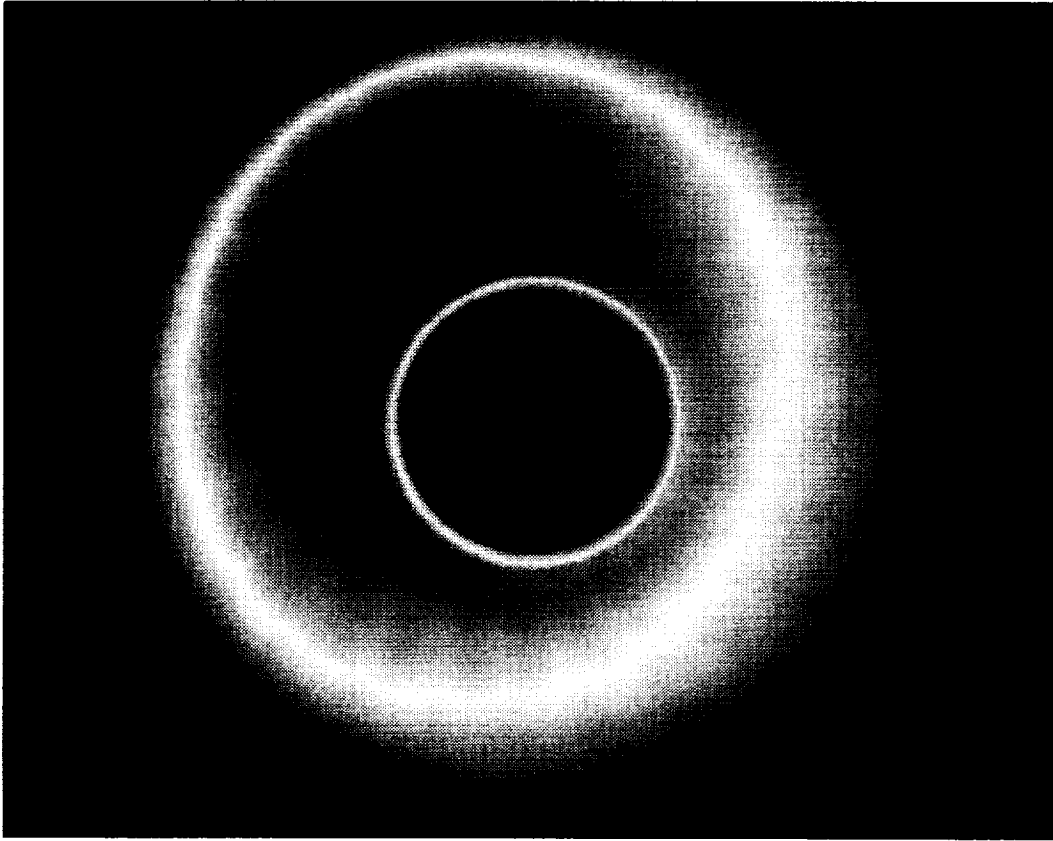


Figure 10.11: Predicted instantaneous near fan face inlet total pressure contours for the AE3007 engine simulation (Mach=0.2, angle of attack=30.0 degrees).

about an axis aligned with the angled inflow. This is a result of the upstream interaction of the fan with the inlet distortion, previously discussed in Chapter 2. The asymmetric loading of the fan resulting from total pressure distortion and local swirl flow effects results in a side to side bias of the inflow, causing the observed asymmetry.

Both the total pressure distortion and local swirl flow distortions cause circumferential variations in the rotor exit total pressure distribution about the circumference of the engine. An illustration of the instantaneous rotor exit total pressure contour pattern is given in Figure 10.12.

The most common method for predicting the effects of inlet distortion on compressor performance is the compressors in parallel method (see e.g. [83]). This technique assumes that, in the presence of inlet distortion, each circumferential "segment" of the compressor operates as if it were independent of the other "segments" of the compressor. All compressor segments behave according to a common operating characteristic. The local inflow properties (defined by the distortion pattern) define the local compressor segment operating condition, and all segments achieve the same exit static pressure. Reasonable qualitative agreement has been achieved with the compressors in parallel methodology. The fan exit total pressure contour plane given in Figure 10.12 supports the basis of the compressors in parallel theory. The exit total pressure pattern is essentially nonmoving in time, which indicates that the fan tends to operate on the local "segment" of the inflow distortion pattern without significant unsteady effects.

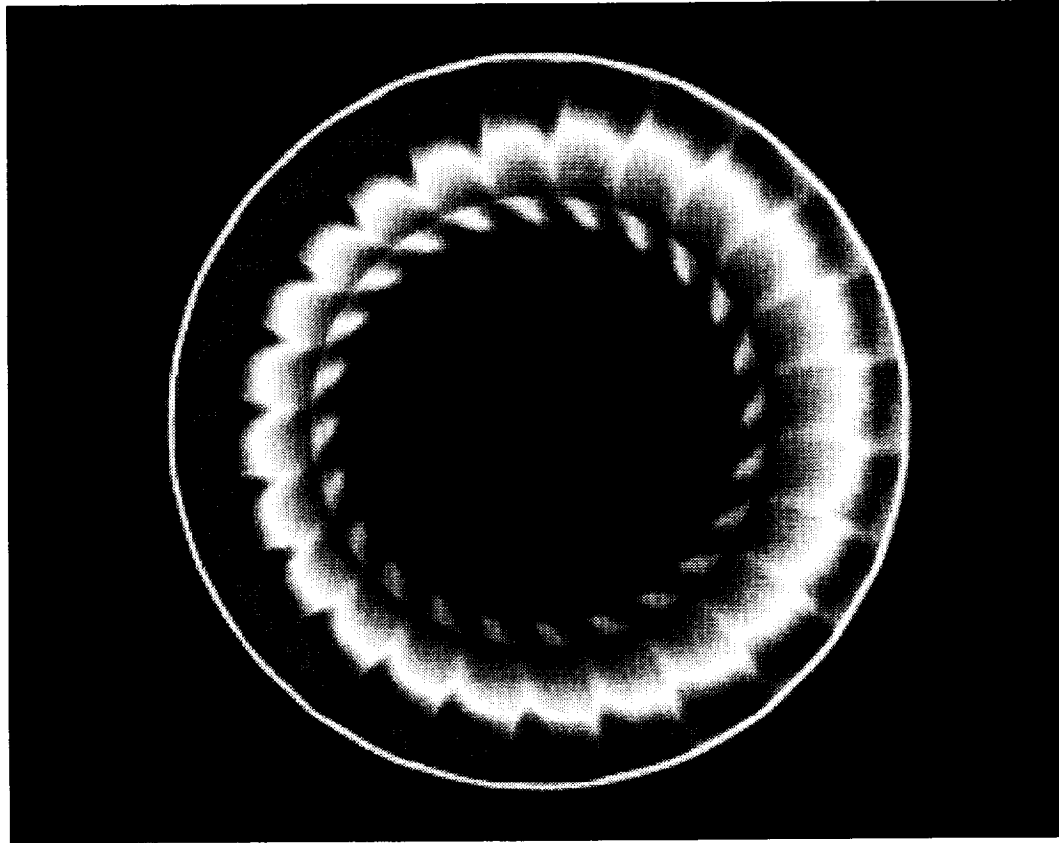


Figure 10.12: Predicted instantaneous near fan face inlet total pressure contours for the AE3007 engine simulation (Mach=0.2, angle of attack=30.0 degrees).

Due to the rather large computational expense associated with calculations of this magnitude, only a single operating point was analyzed with this model; however, it is apparent that a significant number of observations derived from the results of this analysis are consistent with observed phenomena for compressors and turbofan engines. As high power, low cost computer processors become more readily available, detailed configuration studies may ultimately be performed using this analytical technique.

Chapter 11

CONCLUSIONS

Detailed aerodynamic predictions were obtained for various compressor endwall treatment configurations using an advanced 3-D Navier-Stokes analysis technique. Numerical boundary conditions developed include a direct couple scheme for circumferential groove endwall treatments, an endwall treatment time-average approach for predicting steady state flows for discrete slotted endwall treatments, and a time-accurate interface procedure for predicting time-dependent airfoil/endwall treatment aerodynamic interactions.

The analysis of the MIT stator demonstrated the stability of the endwall treatment time-average boundary condition, but the numerical results indicated that for this low speed flow, the beneficial effects of the treatment were somewhat underpredicted by the analysis. This discrepancy was thought to be due to the circumferential averaging procedure used in the numerical approximation.

The analysis was subsequently applied to a high speed flow case utilizing the NASA Rotor 5 geometry with skewed axial slot casing treatment. A detailed analysis of these results indicated that the numerical representation of the discrete treatments provided an adequate increase in total temperature (representative of energy input), but not in total pressure (representative of efficiency).

A detailed design configuration study was performed for the AE3007 fan rotor with circumferential groove, axial slot, blade angle slot, and embedded vane endwall treatments. The results of the study indicated that circumferential grooves should be used as a conservative approach (minimal loss in efficiency) to enhance stall margin, and that additional stall margin improvement would most effectively be afforded by axial slots.

A time-dependent analysis of the NASA Rotor 5 geometry with skewed axial slot casing treatment clearly illustrated the intermittent, time-dependent nature of the recirculating flow pattern for discrete slot endwall treatments. The effect of the flow removal/injection process was related to that of a "booster stage" with the resultant effect of providing additional energy input for the high loss, high blockage clearance vortex and near rotor tip flow.

A time-dependent solution of the aerodynamic interaction between the AE3007 fan rotor and an embedded vane casing treatment with a 1:2 rotor blade to treatment vane airfoil ratio was performed at the untreated rotor near stall condition. Time-averaged results from the unsteady analysis showed favorable agreement with steady state results using the endwall treatment time-average boundary condition. The time-dependent analysis indicated a time-dependent periodic vortical flow structure near the rotor tip, although this was admittedly a rather extreme geometric departure from the original design.

The discrepancies between the endwall treatment time-average predictions and the available experimental data imply that additional developments are required for this type of boundary condition before accurate design analyses can be performed. It is clear that *some* of the beneficial effects of the discrete slot endwall treatments result from a truly time-dependent flow structure, in which case only the time-accurate solution scheme can be expected to accurately capture this flow consistently.

The advanced two-equation ($k - \mathcal{R}$) turbulence model was implemented into the *ADPAC07* analysis and was found to provide superior predictive capability for a shock/boundary layer interaction induced separated flow test case when compared to the existing algebraic (Baldwin-Lomax) turbulence model. The ($k - \mathcal{R}$) model was relatively well-behaved, although solution times are still considered too large for practical production runs.

The iterative implicit algorithm was successfully added to the *ADPAC07* code and demonstrated for a number of flows. Significant improvement in the analysis was observed when a more “fully implicit” formulation was employed. The algorithm in its present state is considered unconditionally stable (based on linear stability analysis), but was found to demonstrate unstable behavior under extreme flow conditions. The stability of this analysis requires further study prior to widespread use.

The implementation of non-reflecting boundary conditions in the *ADPAC07* analysis was relatively straightforward, and several test cases were employed to demonstrate the usefulness of the non-reflecting boundary concept.

Perhaps the most successful aspect of this study was the code parallelization effort and the application of domain decomposition parallel computing using workstation clusters as a virtual parallel computer. This feature of the code was used consistently to solve several rather large computational problems with minimal computational hardware investment.

Computational results for the AE3007 fan rotor both with and without endwall treatment and with and without tip radial distortion were correlated with available experimental test data. The effects of tip radial distortion on rotor spanwise flow properties was accurately captured. The beneficial effects of the circumferential groove endwall treatment were directly related to the reduced blockage near the endwall due to vortex segmentation and blade passage shock repositioning.

Time dependent predictions were performed for the AE3007 fan rotor both with and without circumferential groove endwall treatment for a harmonic circumferential distortion pattern. The computational results verified many assumptions utilized in parallel compressor models of circumferential distortion for compressors systems. The apparent beneficial effects of the circumferential groove endwall treatments were thought to be due to essentially steady state flow physics.

Finally, a time-dependent analysis of the AE3007 turbofan engine operating with inlet distortion due to angle of attack was performed. The analysis included both internal and external flow, and geometric models of the bypass vane and core inlet guide vane were incorporated. The calculations demonstrated several interesting features of realistic engine inlet distortion patterns including fan/distortion aerodynamic interaction, and local incidence effects due to angled inflow on rotor performance.

Bibliography

- [1] Moore, R. D., Kovich, G., and Blade, R. J., "Effect of Casing Treatment on Overall and Blade-Element Performance of a Compressor Rotor," NASA TN D-6538, 1971.
- [2] Osborn, W. M., Lewis, G. W., and Heidelberg, L. J., "Effect of Several Porous Casing Treatments on Stall Limit and on Overall Performance of an Axial Compressor Rotor," NASA TN D-6537, 1971.
- [3] Prince, D. C., Jr., Wisler, D. C., and Hilvers, D. E., "Study of Casing Treatment Stall Margin Improvement Phenomena," NASA CR-134552, 1974.
- [4] Fujita, H., and Takata, H., "A Study of Configurations of Casing Treatment for Axial Flow Compressors," Bull. of JSME, Vol. 27, No. 230, 1985, pp. 1675-1681.
- [5] Takata, H., and Tsukuda, Y., "Stall Margin Improvement by Casing Treatment - Its Mechanism and Effectiveness," ASME Journal of Engineering for Power, Vol. 99, 1977, pp. 121-133.
- [6] Smith, G. D. J., and Cumpsty, N. A., "Flow Phenomena in Compressor Casing Treatment," ASME Journal of Engineering for Gas Turbines and Power, Vol. 107, 1985, pp. 532-541.
- [7] Cheng, P., Prell, M. E., Greitzer, E. M., and Tan, C. S., "Effects of Compressor Hub Treatment on Stator Stall Margin and Performance," Journal of Aircraft, Vol. 21, No. 7, 1984, pp. 469-475.
- [8] Johnson, M. C., and Greitzer, E. M., "Effects of Slotted Hub and Casing Treatments on Compressor Endwall Flow Fields," ASME Journal of Turbomachinery, Vol. 109, 1987, pp. 380-387.
- [9] Lee, N. K. W., and Greitzer, E. M., "Effects of Endwall Suction and Blowing on Compressor Stability Enhancement," ASME Journal of Turbomachinery, Vol. 112, 1990, pp. 133-144.
- [10] Ivanov, S. K., Axial Blower, U. S. Patent No. 3189260, 1965.
- [11] Miyake, Y., Noji, I., Nishikawa, Y., and Murata, S., "The Performance Characteristic of an Air-separator of an Axial Flow Fan," Bull. of JSME, Vol. 28, No. 242, 1985, pp. 1659-1666.
- [12] Azimian, A. R., McKenzie, A. B., and Elder, R. L., "A Tip Treatment for Axial Flow Fans and Compressors," IMechE, April 1987.
- [13] Azimian, A. R., Elder, R. L., and McKenzie, A. B., "Application of Recess Vaned Casing Treatment to Axial Flow Fans," ASME 89-GT-68.

- [14] Crook, A. J., Greitzer, E. M., Tan, C. S., and Adamczyk, J. J., "Numerical Simulation of Compressor Endwall and Casing Treatment Flow Phenomena," ASME Journal of Turbomachinery, Vol. 115, No. 3, pp. 501-512.
- [15] Greitzer, E.M., "The Stability of Pumping Systems- The 1980 Freeman Scholar Lecture," ASME Journal of Fluids Engineering, Vol. 103, June 1981, pp 193-242.
- [16] Greitzer, E.M., "Flow Instabilities in Turbomachines," Lecture Notes-MIT Dept. of Aeronautics and Astronautics, 1991.
- [17] Bowditch, D.N., Coltrin, R.E., "A Survey of Inlet Engine Distortion Capability," NASA TM-83421, prepared for AIAA Joint Propulsion Specialists Conference, June 1983.
- [18] SAE Aerospace Council Division Technical Committee S-16, "Inlet Total-Pressure-Distortion Considerations for Gas-Turbine Engines," SAE Aerospace Information Report AIR-1419, May 1983.
- [19] Mazzawy, R.S., "Multiple Segment Parallel Compressor Model for Circumferential Flow Distortion," ASME Journal of Engineering for Power, April 1977, pp. 288-296.
- [20] Hynes, T.P. and Greitzer, E.M., "A Method for Assessing Effects of Circumferential Flow Distortion on Compressor Stability," ASME Journal of Turbomachinery, Vol. 109, July 1987, pp. 371-379.
- [21] Chue, R., et al., "Calculations of Inlet Distortion-Induced Compressor Flow Field Instability," International Journal of Heat and Fluid Flow, Vol. 10, 1989, pp. 211-223.
- [22] Hall, E. J., Delaney, R. A., and Bettner, J. L., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task I - Ducted Propfan Analysis," NASA CR 185217, NASA Contract NAS3-25270, 1990.
- [23] Hall, E. J. and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task II - Unsteady Ducted Propfan Analysis - Final Report," NASA CR 187106, NASA Contract NAS3-25270, 1992.
- [24] Hall, E. J. and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task V - Counterrotation Ducted Propfan Analysis, Final Report," NASA CR 187126, NASA Contract NAS3-25270, 1992.
- [25] Hall, E. J., Topp, D. A., Heidegger, N. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task VIII - Cooling Flow/Heat Transfer Analysis, Final Report," to be published, NASA Contract NAS3-25270, 1994.
- [26] Hall, E. J., Delaney, R. A., and Bettner, J. L., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task II - Unsteady Ducted Propfan Analysis, Computer Program Users Manual," NASA CR 187105, NASA Contract NAS3-25270, 1991.
- [27] Hall, E. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task V -

- Unsteady Counterrotation Ducted Propfan Analysis - Computer Program Users Manual." NASA CR 187125, NASA Contract NAS3-25270, 1993.
- [28] Hall, E. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task VII - ADPAC User's Manual," NASA Contract NAS3-25270, NASA CR 195472, May, 1996.
 - [29] Hall, E. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task VIII - Film Cooling/Heat Transfer Analysis - Computer Program Users Manual," NASA CR 195360, NASA Contract NAS3-25270, 1994.
 - [30] Barber, T., Choi, D., McNulty, G., Hall, E., and Delaney, R., "Preliminary Findings in Certification of ADPAC," AIAA Paper 94-2240, June, 1994.
 - [31] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.
 - [32] Adamczyk, J. J., Celestina, M. L., Beach, T. A., and Barnett, M., "Simulation of Three-Dimensional Viscous Flow Within a Multistage Turbine," ASME Paper 89-GT-152, 1989.
 - [33] Jorgensen, P. C. E., and Chima, R. V., "An Unconditionally Stable Runge-Kutta Method for Unsteady Flows," NASA TM 101347, 1989.
 - [34] Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, 1978.
 - [35] Goyal, R. K., and Dawes, W. N., "A Comparison of the Measured and Predicted Flowfield in a Modern Fan-Bypass Configuration," ASME Paper 92-GT-298, 1992.
 - [36] Crook, A. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task III-Advanced Fan Section Grid Generator Final Report and Computer Program User's Manual," NASA CR-187129, 1991.
 - [37] Adamczyk, J. J., Celestina, M. L., and Greitzer, E. M., "The Role of Tip Clearance in High-Speed Fan Stall," ASME Paper 91-GT-83, 1991.
 - [38] Moore, R., Private communication.
 - [39] Snyder, R. W., and Blade, R. J., "Analytical Study of Effect of Casing Treatment on Performance of a Multistage Compressor," NASA TN D-6917, 1972.
 - [40] Hall, E. J., Topp, D. A., Heidegger, N. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task VII - Endwall Treatment Inlet Flow Distortion Analysis Final Report," NASA Contract NAS3-25270, NASA CR-195468, July, 1995.
 - [41] Boussinesq, J., "Théorie de l'Écoulement Tourbillant," *Mem. Présentés par Divers Savants Acad. Sci. Inst. Fr.*, Vol. 23, pp. 46-50, 1877.
 - [42] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., "Computational Fluid Mechanics and Heat Transfer," McGraw-Hill, New York, New York, 1984.
 - [43] Hung, C. M., and Kordulla, W., "A Time-Split Finite Volume Algorithm for Three-Dimensional Flow-Field Simulation," AIAA Paper 83-1957, 1983.

- [44] Radespiel, R., Rossow, C., and Swanson, R. C., "Efficient Cell Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations," *AIAA Journal*, Vol. 28, No. 8, pp. 1464-1472, 1990.
- [45] Kreiss, H. O., "Initial Boundary Value Problems for Hyperbolic Systems," *Communications on Pure and Applied Mathematics*, Vol. 23, pp. 277-298, 1970.
- [46] Engquist, B., and Majda, A., "Absorbing Boundary Conditions for the Numerical Simulation of Waves," *Mathematics of Computation*, Vol. 31, pp. 629-651, 1977.
- [47] Erdos, J. I., Alzner, E., and McNally, W., "Numerical Simulation of Periodic Transonic Flow Through a Fan Stage," *AIAA Journal*, Vol. 15, pp. 1559-1568, 1977.
- [48] Giles, M. B., "Nonreflecting Boundary Conditions for Euler Equation Calculations," *AIAA Journal* Vol. 28, No. 12, pp. 2050-2058, 1990.
- [49] Saxer, A. P., "A Numerical Analysis of 3-D Inviscid Stator/Rotor Interactions Using Non-Reflecting Boundary Conditions," MIT GTL Report 209, 1992.
- [50] Martinelli, L., "Calculation of Viscous Flows with a Multigrid Method," Ph. D. Dissertation, MAE Department, Princeton University, 1987.
- [51] Hollanders, H., Lerat, A., and Peyret, R., "Three-Dimensional Calculation of Transonic Viscous Flows by an Implicit Method," *AIAA Journal*, 23, pp. 1670-1678, 1985.
- [52] Rao, K. V., and Delaney, R. A., 1990, "Investigation of Unsteady Flow Through a Transonic Turbine Stage: Part I- Analysis," *AIAA Paper* 90-2408.
- [53] Rosenfeld, M., "Grid Refinement Study of Time Periodic Flow over a Circular Cylinder," *AIAA Paper* 93-0433, 1993.
- [54] Rogers, S. E., and Kwak, D., "An Upwind Differencing Scheme for the Time-Accurate Incompressible Navier- Stokes Equations," *AIAA Paper* 88-2583, 1988.
- [55] Rosenfeld, M., Kwak, D., and Vinokur, M., "A Solution Method for the Unsteady and Incompressible Navier Stokes Equations in Generalized Coordinate Syaytems," *AIAA Paper* 88-0718, 1988.
- [56] Lecointe, Y., and Piquet, J., "On the Use of Several Compact Methods for the Study of Unsteady Incompressible Viscous Flow Round a Circular Cylinder," *Computers and Fluids*, Volume 12, No. 4, pp. 255-280, 1984.
- [57] Martinez, G., "Caracteristiques Dynamiques et Thermiques de l'Ecoulements Atour d'un Cylindre Circulaire a Nombres de Reynolds Moderes," These D. L. - I. N. P. Toulouse, 1979.
- [58] Lin, C. L., Pepper, D. W., and Lee, S. C., "Numerical Methods for Separated Flow Solutions Around a Circular Cylinder," *AIAA Journal*, Volume 14, pp. 900-907, 1976.
- [59] Thoman, D., and Szewczyk, A., "Time-Dependent Viscous Flow Over a Circular Cylinder," *Physics of Fluids Supplement II*, pp. 79-86, 1969.
- [60] Wille, R., "Karman Vortex Streets," *Advances in Applied Mechanics*, Volume 6, pp. 273, 1960.

- [61] Kovasznay, L. S. G., "Hot Wire Investigation of the Wake Behind Cylinders at Low Reynolds Numbers," *Proceedings of the Royal Society of Aeronautics*, Volume 198, pp. 174-190, 1949.
- [62] Roshko, A., "On the Development of Turbulent Wakes from Vortex Streets," NACA Report 1191, 1954.
- [63] Wilcox, D. C., "Turbulence Modeling for CFD," DCW Industries, La Cañada, California, 1993.
- [64] Spalding, D. P., "Monograph on Turbulent Boundary Layer," Imperial College, Mechanical Engineering Department Report TWF/TN/33, 1976.
- [65] Wolfshtein, M. W., "The Velocity and Temperature Distribution in One Dimensional Flow with Turbulence Augmentation and Pressure Gradient," *International Journal of Heat and Mass Transfer*, Vol 12, p. 301, 1969.
- [66] Patankar, S. V., and Spalding, D. B., "Heat and Mass Transfers in Boundary Layers," 2nd Edition, Intertext Books, London, 1970.
- [67] Launder, B. E., and Spalding, D. B., "The Numerical Computation of Turbulent Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 3, pp.269-289, 1974.
- [68] Goldberg, U., "Towards a Pointwise Turbulence Model for Wall-Bounded and Free Shear Flows," *Boundary Layer and Free Shear Flows*, ASME FED-Vol. 181, pp. 113-118, 1994.
- [69] Baldwin, B. S., and Barth, T. J., "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," AIAA Paper 91-0610, 1991.
- [70] Launder, B. E., and Sharma, B. I., "Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disk," *Letters in Heat and Mass Transfer*, Vol. 1, pp. 131-138, 1974.
- [71] Bachalo, W. D., and Johnson, D. A., "An Investigation of Transonic Turbulent Boundary Layer Separation Generated on an Axisymmetric Flow Model," AIAA Paper 79-1479, 1979.
- [72] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 91-1596, 1991.
- [73] Arnone, A. A., Liou, M. S., and Povinelli, L. A., "Multigrid Time-Accurate Integration of Navier-Stokes Equations," AIAA Paper 93-3361-CP, 1993
- [74] Arnone, A., Pacciani, R., and Sestini, A., "Multigrid Computations of Unsteady Rotor-Stator Interaction Using the Navier-Stokes Equations," Submitted for Presentation to the 1995 ASME Gas Turbine Conference, 1995.
- [75] Melson, N. D., Sabetrik, M. D., and Atkins, H. L., "Time-Accurate Navier-Stokes Calculations with Multigrid Acceleration," Presented at the Sixth Copper Mountain Conference on Multigrid Methods, Copper Mountain, Colorado, April 4-9, 1993.
- [76] Quealy, A., Cole, G. L., and Blech, R. A., "Portable Programming on Parallel/Networked Computers Using the Application Portable Parallel Library (APPL)," NASA TM-106238, 1993.
- [77] Zukauskas, A., and Slanciauskas, A., "Heat Transfer in Turbulent FLuid Flows," Hemisphere Publishing Corporation, New York, New York, 1987.

- [78] Hylton, L. D., Mihelc, M. S., Turner, E. R., Nealy, D. A., and York, R. E., "Analytical and Experimental Evaluation of the Heat Transfer Distribution Over the Surfaces of Turbine Vanes," NASA CR-168015, 1983.
- [79] White, F. M., "Viscous Fluid Flow," McGraw-Hill Publishing, New York, New York, 1974.
- [80] Delaney, R. A., Helton, D. J., Bennet, W. A., Dunn, M. G., Rao, K. V., and Kwon, O., "Turbine Vane-Blade Interaction", WRDC-TR-89-2154, Vol. I, March, 1990.
- [81] Mulac, R. A., "A Multistage Mesh Generator for Solving the Average-Passage Equation System," NASA CR-179539, 1988.
- [82] Longley, J. P., "Inlet Distortion and Compressor Stability", Ph. D. Dissertation, Cambridge University Engineering Department, 1988.
- [83] Cumpsty, J. P., "Compressor Aerodynamics", Wiley and Sons Publishing, New York, New York, 1989.

Appendix A

ADPAC07 Utility Program Developments

Several *ADPAC07* utility programs are described in this section. These programs were developed during the course of this study to simplify the application of the *ADPAC07* code for many complex problems.

A.1 *PATCHFINDER* Utility Program

An *ADPAC* utility program was developed to aid in the construction of boundary condition files for complex, interconnected multiple block mesh systems. The new utility, named *PATCHFINDER*, reads in an *ADPAC* mesh (a *PLOT3D* binary multiple-block grid) and determines which blocks share matching faces. As the grid is read, the faces are striped off into a separate array. Individual points on these faces are compared until a “point” match is found. Neighboring points are then compared to find a “cell” match. This also determines the relative directions of the matching indicies. From this cell match, a common face area is swept out and a **PATCH** boundary condition statement is written using the bounding indicies of the common face area.

After all the **PATCH** specifications have been written, any remaining surfaces not accounted for will have a solid surface wall (**SSVI**) boundary condition written out. This allows the user to simply replace a few solid wall statements with the proper inlet and exit boundary conditions and start running. The *PATCHFINDER* user input file contains approximately ten variables to customize a *PATCHFINDER* run to each grid although many grids can be processed without special input. Since the majority of boundary conditions prescribed for most geometries are block patches and solid walls, running *PATCHFINDER* will greatly simplify the generation of a boundary data file.

Several methods were implemented to accelerate the *PATCHFINDER* search and compare process. Some of these methods include using multi-grid and bounding cube limits. If a mesh is created to be run with *ADPAC* using multi-grid, this can also be used to accelerate *PATCHFINDER* since boundary conditions must be consistent across multi-grid levels. Each level of multi-grid decreases the number of face points by a factor of 4; therefore, with 3 levels of multi-grid a decrease in run time of roughly sixteen times can be expected. The bounding cube limit method creates a limiting cube enclosing all the cell centers of a block face. Before any individual face points are checked, the corresponding bounding cubes are checked for intersection. If the cubes do not intersect, then no matching points are possible;

Test Case Grid	Grid Points	Block Faces	Multi-Grid Levels	PATCH(es) Written	Elapsed Time
2-D Axisymmetric Seal Cavity (Phadke/Owen #5)	5,438	36	3	12	0:02
O-Grid Capped with H-Grids	34,595	18	1	16	11:14
Combustion Can	160,040	72	3	72	0:07
Exhaust Mixer	197,190	36	2	22	1:23
Vane Blade Interaction	232,645	30	3	30	0:19
Rotor-Stator-Rotor with Seal Cavity Grid (Unsteady, Multiple Pitches)	259,777	258	2	214	1:05
Airplane	483,876	618	1	808	16:31
AE3007 Fan with 5-Groove Casing Treatment	642,479	21	3	50	1:46
Rotor-Stator-Rotor with Seal Cavity Grid	715,001	90	3	62	1:44

Figure A.1: Approximate wall clock run times for various *PATCHFINDER* test case configurations run on a Silicon Graphics 4D-35 workstation (5 MFLOP).

this greatly reduces the number of individual point searches.

Several test cases were run using *PATCHFINDER*. These included both 2-D and 3-D geometries with and without multi-grid. Each of these runs was timed to evaluate the efficiency of the program. The resulting approximate wall clock times are shown in the table in Figure A.1. All cases were run on the same machine under similar conditions so that relative comparisons can be made. Most of the test cases using multi-grid finished in under two minutes. One of the largest and most complex test cases was based on a mesh with over 480,000 points and over 800 matching faces. *PATCHFINDER* was able to correctly identify all the patches in under 17 minutes (on a Silicon Graphics 4D-35 workstation) as opposed to the approximately two days required to correctly specify the **PATCH** connections by hand. The *PATCHFINDER* time was limited in this case since the grid contained only one level of multi-grid. From the times recorded in the table, the advantage of using multi-grid becomes apparent.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1996	3. REPORT TYPE AND DATES COVERED Final Contractor Report		
4. TITLE AND SUBTITLE Task 7—Endwall Treatment Inlet Flow Distortion Analysis Final Report		5. FUNDING NUMBERS WU-5-03-11 C-NAS3-25270		
6. AUTHOR(S) E.J. Hall, D.A. Topp, N.J. Heidegger, G.S. McNulty, K.F. Weber, and R.A. Delaney				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Allison Engine Company 2001 S. Tibbs/P.O. Box 420 Indianapolis, Indiana 46206-0420		8. PERFORMING ORGANIZATION REPORT NUMBER E-9647		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-195468		
11. SUPPLEMENTARY NOTES Project Manager, Christopher J. Miller, Propulsion Systems Division, NASA Lewis Research Center, organization code 2770, (216) 433-6179.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 07 This publication is available from the NASA Center for Aerospace Information, (301) 621-0390.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) The overall objective of this study was to develop a 3-D numerical analysis for compressor casing treatment flowfields, and to perform a series of detailed numerical predictions to assess the effectiveness of various endwall treatments for enhancing the efficiency and stall margin of modern high speed fan rotors. Particular attention was given to examining the effectiveness of endwall treatments to counter the undesirable effects of inflow distortion. Calculations were performed using three different gridding techniques based on the type of casing treatment being tested and the level of complexity desired in the analysis. In each case, the casing treatment itself is modeled as a discrete object in the overall analysis, and the flow through the casing treatment is determined as part of the solution. A series of calculations were performed for both treated and untreated modern fan rotors both with and without inflow distortion. The effectiveness of the various treatments were quantified, and several physical mechanisms by which the effectiveness of endwall treatments is achieved are discussed. This report represents the cumulative efforts of Tasks 6 and 7 under NASA Contract NAS3-25270.				
14. SUBJECT TERMS Fluid dynamics stall		15. NUMBER OF PAGES 220		
		16. PRICE CODE A10		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

National Aeronautics and
Space Administration

Lewis Research Center
21000 Brookpark Rd.
Cleveland, OH 44135-3191

Official Business
Penalty for Private Use \$300

POSTMASTER: If Undeliverable — Do Not Return